Forecasting from Physiological Time Series Through Supervised Learning

by

Shamsul Kabir Masum

The thesis is submitted in partial fulfilment of the requirements for the award of the degree of Doctor Philosophy of the University of Portsmouth.

August, 2019

Declaration

This thesis has been written under the supervision of Dr John Chiverton, Dr Ying Liu and Dr Branislav Vuksanovic. While registered as a candidate for the above degree; I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

.....

Shamsul Kabir Masum

Abstract

Time series analysis of physiological data could potentially help healthcare professionals in decision making and intervening dangerous clinical events. Time series analysis of physiological data have been mostly limited to classification based approaches. This might be where only a single point or a class is predicted. Classification based approaches are useful in themselves but they are unable to provide any detail and consequently any insight in to a clinical event. In contrast to this, accurate forecasting of physiological time series data is an open challenge and this could help to provide more details regarding the conditions that might lead to a dangerous clinical event. The few existing works that forecast physiological data have not considered a comparison of forecast strategies and data approaches. Researchers have previously found for other disciplines such comparisons can help in building an efficient model. Moreover, the forecast outcome was not applied to any medical applications. Machine learning algorithms and methodologies have received little attention and are yet to be used in a detailed exploration of time series forecasting for physiological data. This thesis aims to explore the scope of time series forecasting of physiological data for the early intervention of dangerous clinical events (hypotension and bradycardia) through machine learning. Different forecasting strategies and data approaches are considered here and analysed in combination with different machine learning algorithms to find the best. Forecast models are built to explore the scope of different machine learning techniques for the intervention of dangerous clinical events. Patients' physiological time series data (blood pressure and heart rate) are used from the MIMIC II and III databases.

The findings of this study appear to demonstrate that forecasting of physiological time series data can be used for the intervention of dangerous clinical events. It is found that MIMO and DIRMO are the two best strategies and the multivariate based approaches appear to outperform the univariate approaches in forecasting physiological data. The work then goes on to show that supervised machine learning for regression appears to predict events with greater accuracy than without regression. It is also observed that the forecast model with a regression algorithm performs better when forecasting in the gap window as well as the target window. This is only true for a gap window of up to 30 minutes. A forecast model with no gap window forecasts events with more than 99 per cent accuracy. All these scenarios could potentially be informative and useful to health care professionals, forecast model developers and researchers. Furthermore, the availability of the forecast regression data could be a further level of potentially useful information to a healthcare professional.

Acknowledgements

First and foremost, all praises and thanks to Allah - God Almighty.

I owe a tremendous debt of gratitude to Dr John Chiverton, Dr Ying Liu and Dr Branislav Vuksanovic for their guidance and support throughout the entire PhD journey. They all have been a great teacher, mentor, and a role model that I will follow in my future academic career.

I would especially like to thank Dr Mohamed Bader and Mr Simarjot Dahella from School of Computing for their suggestion and guidance in extracting physiological data from MIMIC database. I would also like to thank Dr Mohamed G. Hassan-Sayed for his valuable suggestions during the yearly review process. Moreover, special thanks to Dr Matt Parker from the School of Pharmacy and Biomedical Sciences for discussing the vital sign and its importance in medical applications. I want to thank the external and internal examiners, Dr Zina Ibrahim and Dr Linda Yang, for their feedback and suggestions, which helped me improve the thesis's presentation.

Special thanks go to Linda Janes and all staff of the international office for helping me out with administrative issues throughout my PhD journey. I want to thank all the staff of the School of Engineering for your support. To all my friends and all my fellow PhD students in the department. Thank you for your support and friendship.

Most importantly, I want to thank my family members, especially my late dad, who taught me the value of education. My parents have made many sacrifices to put me in a position where I am today.

Contents

List of Figures

List of Tables

List of Abbreviations

ABP Arterial Blood Pressure

AI Artificial Intelligence

ARIMA AutoRegressive Integrated Moving Average

AUROC Area Under ROC Curve

BI − LSTM Bidirectional- Long-Short-Term-Memory

BP Blood Pressure

BPM Beats Per Minute

CNN Convolutional Neural Network

CRF Conditional Random Field

DBP Diastolic BP

DL Deep Learning

ECG ElectroCardiogram

EEG ElectroEncephalogram

EHR Electronic Health Record

EPMA European Association for Predictive, Preventive and Personalised Medicine

HE Hypotensive Episode

HMM Hidden Markov Model

HR Heart Rate

ICU Intensive Care Unit

KNN k Nearest Neighbor

LR Logistic Regression

LSTM Long-Short-Term-Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MIMIC Multiparameter Intelligent Monitoring in Intensive Care

ML Machine Learning

MLP Multilayer Perceptron

MSE Mean Squared Error

NN Neural Network

NPV Negative Predictive Value

PICU Paediatric Intensive Care Unit

PPV Positive Predictive Value

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

RR Respiratory Rate

SBP Systolic BP

List of Publications and Presentations

- Conference Papers:

  1) Masum, S., Liu, Y., and Chiverton, J. (2017, July). Comparative analysis of the outcomes of differing time series forecasting strategies. In 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 1964-1968). IEEE.

  2) Masum, S., Liu, Y., and Chiverton, J. (2018, June). Multi-step time series forecasting of electric load using machine learning models. In International Conference on Artificial Intelligence and Soft Computing (pp. 148-159). Springer, Cham.

  3) Masum, S., Chiverton, J. P., Liu, Y., and Vuksanovic, B. (2019, December). Investigation of Machine Learning Techniques in Forecasting of Blood Pressure Time Series Data. In International Conference on Innovative Techniques and Applications of Artificial Intelligence (pp. 269-282). Springer, Cham.

- Journals:

  1) Masum, S., Chiverton, J., Liu, Y. and Vuksanovic, B. (2020, January). Machine Learning in Intervening Clinical Events Analysing Physiological Time Series Data. Journal of Biomedical Informatics (Under Review).

- Poster Presentations:

  1) Masum, S., Chiverton, J., Liu, Y. and Vuksanovic, B. (2019, June). Investigation of Machine Learning Techniques in Forecasting of Physiological Time Series Data. In PHT-UoP Health and Technology Research and Innovation Conference.

Chapter 1

Introduction

This chapter discusses the motivation that has driven the work in this thesis. It mentions the research gaps. In particular, it discusses the motivation for the experiments that help contribute to filling the research gaps. Descriptions of the contributions then follow. Finally, the organisation of the thesis and contents of individual chapters are summarised.

1.1 Motivation

Physiological signals, also known as vital signs, reflect the physiological state of the patient. Collections of various vital signs can be used to describe the underlying state of an individual. Primary physiological data include temperature, heart rate, blood pressure and respiratory rate. These are frequently measured. This is because they can be used to define a patients' wellbeing status. In particular they can help diagnose and determine the seriousness of health conditions and contribute in the treatment process [Kwon et al., 2018, Xue et al., 2018, Ghosh et al., 2016, Lee and Mark, 2010b, Angner et al., 2009, Ghosh et al., 2017]. Most importantly, physiological signals can play a pivotal role in a proactive healthcare model. They can potentially help in predicting a deteriorating physical condition of an individual. Lack of such critical vital sign information may cause delayed diagnoses or misjudgement of the status of an illness or injury [Elliott and Coventry, 2012]. Measurement and identification of vital signs depends on several factors such as race, age, sex, anxiety, pain, stress,

mental condition and environmental condition of the individual. There is a generally accepted vital sign range as shown in Table 1.1 [Subbe et al., 2001]. The modified early warning score system is used as a tool by healthcare professionals in identifying patients' with declining conditions. Such a tool considers subtle changes in a number of different physiological data, it also helps to consider whether there are any significant changes for a single vital sign. A healthcare professional provides the usual care and observation to a patient with a low early warning score. Whereas, a patient with a high early warning score is observed by a healthcare professional and transferred to a special care unit if required. An early warning score of 5 or more suggests that the patient is in a critical condition and should be transferred to an intensive care unit otherwise death is likely to occur. The ranges of physiological data and scores are calibrated to different populations.

Table 1.1: Modified early warning score

	Score						
Parameters	3	2	1	0	1	2	3
Systolic Blood pressure (mmHg)	<70	71–80	81–100	101–199	-	≥ 200	-
Heart rate (bpm)	-	<40	41–50	51–100	101–110	111–129	≥ 130
Respiratory rate (bpm)	-	<9	-	9–14	15–20	21–29	≥ 30
Temperature (°C)	-	<35	-	35–38.4	-	≥ 38.5	-

A patients' physiological status can deteriorate suddenly without any prior notification. When this happens, the current healthcare paradigm requires that a healthcare professional acts swiftly to save the patient. This paradigm is referred to as a 'reactive patient care paradigm'. Existing early warning systems consist of predefined rules. These are applied to vital signs to generate a reactive alarm, and in some cases, cause a significant number of false alarms. Researchers have found that there have been many cases where patients' have received incorrect diagnoses and treatments [McGlynn et al., 2003] [Pinsky, 2007] [Ghosh et al., 2016]. Moreover, current systems operate on aggregate rules of physiological data and do not take advantage of sequential information. Systems directly map the input physiological signals to output values without explicitly modelling the underlying temporal dependencies [Su et al., 2018]. On the other hand, it is almost impossible for a human clinician to analyse the continuous and

quantitative complex medical data in practical clinical settings. Furthermore, there is a huge gap between research and clinical practice [Bennett and Hauser, 2013]. The European Association for Predictive, Preventive and Personalised Medicine (EPMA) has criticised the current health care model around Europe and worldwide. EPMA has motivated its representatives to explore advanced research in the healthcare area to come up with solutions to overcome of the current reactive nature of healthcare [Golubnitschaja et al., 2014].

Advancements in medical technologies are allowing us to capture vital sign data from patients' in intensive care units (ICU) with time-varying phenomena. ICU at hospitals are considered the best place for a wealth of patients' data collected from many advanced specialised units [Johnson et al., 2016a] and includes vital signs, events, images, waveforms, laboratory results and much more. Unfortunately, most of the captured data has remained underutilised and the healthcare sector is very often unable to take full advantage of it. Captured data can be processed for classification and regression scenarios. Temporal aspects of the data can also be considered. Given this wealth of patient data, a timely question can be raised as to whether it is possible to predict the future of a patients' physiological state. Forecasting a patients' deteriorating physiological signs could help to alert or provide additional insight to healthcare professionals about the patients' state. Such scenarios can help healthcare professionals to intervene in advance according to the situation. This could potentially contribute to a change in the reactive patient care paradigm to help provide more predictive patient care. However, achieving such predictive patient care comes with a number of challenges including access to a rich enough source of data, developing the ability to understand and analyse the data, and developing the ability to learn from the data. Furthermore, it is crucial to explore events such as deterioration events. Then look back from that event to find whether there was any signature left to identify it. Part of this challenge is to build a forecast model that can identify and capture that signature. Such a forecast model can then be deployed to patients' bedsides for further testing.

Time series forecasting has been used in economics and statistics, where linear statistical models such as AutoRegressive Integrated Moving Average (ARIMA) have been used to predict linear data. However, most real-world applications involve nonlinear data; yet nonlinear time series forecasting and analysis lag behind [Tabachnick et al.,

2007,Kam, 2014,Lineesh et al., 2010]. Machine learning models are data-driven models which use patterns embeded within the data to build forecasting models; they deal with nonlinear data by determining the stochastic dependencies between a set of input and output variables of the past data. Machine learning has shown potential in healthcare especially in decision support [Frizzell et al., 2017, Gulshan et al., 2016, Schaefer et al., 2005]. Machine learning allows for the extraction of features from medical datasets in a way that is efficient for a computer in comparison to a medical expert. Machine learning techniques will likely be used to help solve some of the most complex medical problems [Clifton et al., 2015]. Researchers have used time-independent medical time series data as statistical features for different prediction tasks. This has included ICU physiological instability through decision rules [Eshelman et al., 2008] and hypotension during traumatic brain injury through bayesian artificial neural networks [Donald et al., 2012]. Machine learning has been used for decision making from real-world temporal data. Researchers have considered the temporal dependency of the data in prediction problems. Problems have included: hospitalisation of haemodialysis patients' [Bellazzi et al., 2005]; requirement of mechanical ventilation for greater than 24 hours [Bellazzi et al., 2011]; various illnesses [Patnaik et al., 2011]; and disease progression [Perer and Wang, 2014]. However, in time series, data are typically dependent over time. In such cases, some specialised sequential machine learning techniques would be more suitable rather than traditional machine learning approaches which operate on static data.

Sequential machine learning techniques, such as Recurrent Neural Networks (RNNs), are designed to process time-series sequential data. RNNs have been used in predicting text sequences and for language processing [Sutskever et al., 2014, Graves, 2013]. Long-Short-Term-Memory (LSTM) and Bidirectional- LSTM (BI-LSTM) are extensions of RNNs. They have gained attention in time series forecasting for different fields. An example includes traffic speed prediction [Ma et al., 2015] where, LSTMs outperformed elman neural networks, time delay neural networks, kalman filters, Support Vector Machines (SVMs) and ARIMA . This was also found to be the case for solar power forecasting [Gensler et al., 2016] where LSTM outperformed deep belief network and Multilayer Perceptron (MLP). Also, for electric load forecasting [Masum et al., 2018] where LSTM outperformed ARIMA and natural language processing [Zhou et al., 2016] where LSTM beat SVM, Convolutional Neural Network(CNN) and RNN.

In recent times, LSTM and BI-LSTM have been applied to medical time series data. Lipton et al. evaluated LSTM to recognise patterns in multivariate time series of clinical measurements [Lipton et al., 2015b]. Nguyen et al. used both LSTM and BI-LSTM models to predict mortality outcomes in ICUs from physiological time-series data [Nguyen et al., 2017]. They showed that BI-LSTMs outperformed LSTMs, Logistic Regression (LR) and gated recurrent units. Zhu et al. considered supervised BI-LSTM to predict ICU mortality [Zhu et al., 2018] and found that they outperformed CNN, LSTM and gated recurrent unit. Despite having this attention, these RNNs are yet to achieve superior performance in modelling healthcare time series data. Moreover, machine learning has received little attention in sequence-to-sequence prediction [Sutskever et al., 2014] [Luong et al., 2015] or multi-step ahead forecasting in medical applications. Literature has considered machine learning algorithms mainly in terms of sequence classification [Aggarwal, 2014], which has been limited to single-step forecasting. Sequence-to-sequence prediction predicts an output sequence for a given input sequence. Sequence classification predicts a class or label for a given input sequence. Sequence-to-sequence prediction has the advantage of forecasting the sequence. This is useful for many critical applications, i.e. speech recognition and machine translation, which are best described with sequences [Sutskever et al., 2014]. However, sequence-to-sequence prediction is found to be more challenging than sequence classification where a single next value is predicted in the sequence. In sequence-to-sequence predictions, a problem can be considered as time series forecasting if the input and output sequences are a time series. Multi-step time series forecasting of physiological data with regression algorithms would be of great benefit. They could potentially be used to help to intervene in dangerous clinical events. Moreover, this would aid health care professional in the decision-making process.

1.2 Research Gap

Existing works have considered classification algorithms, which mostly directly map the input physiological signals to output values without modelling the temporal dependencies in underlying physiological data dynamics. The accurate prediction of multi-step forecasts is a particular open challenge. Moreover, machine learning algorithms

and methodologies are yet to fully explore multi-step forecasting scenarios [Weigend, 2018, Masum et al., 2018]. Continuous monitoring, which often requires long horizon forecasting, is an integral part of clinical decision making. This was shown in glucose monitoring [Mclachlan et al., 2007] and EEG monitoring in the ICU [Vespa et al., 1999]. Accurate forecasting of physiological time series data can help a healthcare professionals to intervene early in treating patients'. This can therefore help to contribute to a preventive patient care paradigm.

However, there are few works that actually perform forecasting of physiological time series data, but have not considered different forecast strategies [Lee and Mark, 2010b] [Billis and Bamidis, 2014] [Li et al., 2017] [Sideris et al., 2016] [Su et al., 2018]. Comparison of forecast strategies in these studies would help in selecting the best forecasting strategy for the model, as Ben Taieb et al. showed that forecast strategies play a vital role in forecasting [Taieb et al., 2012]. Moreover, literature is missing the comparison of data distributions (univariate and multivariate) in time series forecasting of physiological data. From the current work, it has been found that researchers have either used univariate or multivariate approaches in forecasting physiological time series data without comparing these approaches [Lee and Mark, 2010b] [Billis and Bamidis, 2014] [Li et al., 2017] [Sideris et al., 2016] [Su et al., 2018]. A comparison between these data approaches would have helped to find the best approach for a specific application. Moreover, no work explores the utilisation of forecast outcomes in medical applications. It has also been found that the lack of collaborations between these different fields is creating a barrier to further developments.

The scope of machine learning algorithms and methodologies is yet to be explored in long-term forecasting in medical applications. Algorithms like naive bayes [Lipton et al., 2015a], support vector machines [Ongenae et al., 2013], support vector regression [Smola and Schölkopf, 2004], gradient boosted regression trees [Friedman, 2001], factorization machine [Rendle, 2010] and Multilayer Perceptron (MLP) are not suitable for temporal data and long-term dependencies. Gaussian processes, Hidden Markov Models (HMMs), CRFs are known as machine learning-based sequential techniques. These models are found to be effective in dealing with temporal data but are not suitable for handling long-term dependencies [Hill et al., 1996, Baccouche et al., 2011, Lafferty et al., 2001]. Long Short-Term Memory (LSTM) overcomes the long-

term dependencies issues [Hochreiter and Schmidhuber, 1997]. However, It has mostly been limited to time series classification when applied to medical applications [Lipton et al., 2015b] [Nguyen et al., 2017] [Zhu et al., 2018] [Carlin et al., 2018] [Nguyen et al., 2017] [Lipton et al., 2015b] [Jo et al., 2017] [Chauhan and Vig, 2015] [Lipton et al., 2015b]. Time series classification only predicts the next value. Predicting multiple values for a medical application could help open up new potential medical applications. Moreover, few works that consider LSTM in the forecasting of physiological data does show that LSTM outperforms traditional machine learning algorithms [Li et al., 2017] [Sideris et al., 2016] [Su et al., 2018]. The algorithms that have considered forecasting of physiological time series data have missed hyperparameter tuning [Lee and Mark, 2010b] [Billis and Bamidis, 2014] [Li et al., 2017] [Sideris et al., 2016] [Su et al., 2018]. Combinations of different machine learning algorithms found effective in forecasting [Kim and Kim, 2019] [Huang and Kuo, 2018] [Kim et al., 2019]. However, the combination of different machine learning algorithms in forecasting physiological time series data is not yet explored.

In specific to intervening critical event, existing works only consider predicting the class of the problem using classification algorithms. However, classification algorithms do not reflect the wide range of these class, i.e. a hypotensive event has a long-range of 0 to 59 in its definition. Following the definition, both 0 and 59 will be considered as a hypotensive event. Whereas, healthcare professionals will consider such ranges differently in terms of actions and treatments. Moreover, from the outcome of classification algorithms, it was found that the gap window plays a vital role in the performance of the model [Lee and Mark, 2010b] [Ghosh et al., 2016]. A gap window separates the observation and target windows. This indicates that observations in the gap window potentially carry valuable information. Analysis of such useful information in the gap window would help to improve the model accuracy. However, a model with a classification algorithm needs to omit this information to create the gap. A methodology that has the ability to utilise the valuable information available in the gap window could help to intervene in the critical events better. Moreover, the unbalanced nature of the data was not considered in the experimented datasets, which could lead to overfitting and biasing problems [Lee and Mark, 2010b] [Ghosh et al., 2016]. The experimented dataset was also not shared in most cases, thus making it harder to compare results.

Therefore a standard dataset would help to solve the issue. Lee and mark [Lee and Mark, 2010b] is the only work that applied regression algorithms to forecast the hypotensive events. However, their work omitted the gap information when using the regression algorithm. Moreover, investigations on predicting bradycardia is very minimal. A few efforts concentrated on risk factors and management of bradycardia, rather than predicting events.

All these factors mentioned have helped motivate the desire to explore the scope of machine learning techniques to physiological time series forecasting and more specifically in intervening dangerous clinical events. Two research problems need to be addressed and investigated. First, the combination of machine learning techniques, which can provide the best performance for forecasting physiological time series data. Second, how the forecast outcome can be utilised to intervene dangerous clinical events.

1.3 Research Aims and Objectives

This thesis aims to evaluate the performance of machine learning-based forecasting of physiological time series data. In particular, potentially as an aid in the early intervention of adverse clinical events. To achieve this aim, the objectives of the research are as follows:

- Compare the implementation of different forecasting strategies using a number of state-of-the-art machine learning algorithms to forecast future values of physiological time series data.

- Evaluate the performance of univariate and multivariate forecasting techniques of physiological time series data.

- Compare the performance of machine learning algorithms in forecasting physiological time series data.

- Explore the scope of machine learning algorithms (classification and regression) in analysing blood pressure and heart rate time series to predict potentially dangerous clinical events.

1.4 Research Contributions

This research addresses the existing research gaps by investigating different machine learning techniques and algorithms in forecasting physiological data. Comparing combinations of different techniques can help to explore the best method for forecasting physiological data. This would help to address and answer the lack and current limitations of forecasting physiological data. Application of the forecast outcomes to help with the intervening of dangerous clinical events would also help to address the limitations of the current existing research literature. Moreover, hyperparameter tuning, comparisons of forecast strategies, data approaches and combinations of algorithms in the model building process, also help to address the gap of existing research.

The main contributions of the research are as follows:

- Investigated forecasting strategies of physiological data 30 minutes in advance. Physiological data (HR and BP) were collected from the hypotension group of an ICD-9 code of MIMIC 2 database. Forecast models were built by combining machine learning algorithms and forecast strategies. This enabled different forecasting strategies to be compared in forecasting physiological time series data a current limitation of the existing literature. This helped to identify the best possible forecast strategies, algorithms and their combination in forecasting BP and HR time series data. Results appear to suggest that MIMO and DIRMO are the two better performing forecast strategies in terms of low RMSE in forecasting BP and HR. This is in comparison to Recursive, Direct, DirRec and RECMO forecast strategies. Results also showed that BI-LSTM machine learning algorithm outperforms ordinary LSTM.

- Investigated univariate and multivariate forecast approaches for time series forecasting of physiological data along with machine learning algorithms. The investigation was conducted to compare different data approaches when applied to physiological time series forecasting a current limitation in the existing research literature. Physiological data (HR and BP) were extracted from the hypotension group of an ICD-9 code of MIMIC 2 database. Forecast models were built by combining machine learning algorithms and forecast strategies. Forecast models aims to explore which data approach (univariate vs multivariate) performs better

in forecasting BP time series 30 minutes in advance. The comparative results appear to show that forecast models consisting of multivariate data (BP and HR) are more reliable in forecasting BP compared to univariate data. Comparative analysis between MIMO and DIRMO forecasting strategies appear to show that multi-step forecasting seems to be more reliable than single-step forecasting for univariate and multivariate cases. Results also appear to show that BI-LSTM RNN is the best model in forecasting blood pressure compared to ordinary LSTM and CNN models.

- BP and HR time series have also been analysed by both classification and regression machine learning algorithms in terms of the hypotensive and bradycardia events. A hybrid model combining two regression algorithms are also presented in predicting events. Findings suggest that it is possible to forecast clinical events with greater accuracy and further in to the future than other existing methods. This was combined with a relatively detailed investigation on machine learning techniques and hyper parameter tuning. This helps to address the current lack of application of forecasting outcomes to medical applications, hyperparameter tuning and combination of machine lerning algorithms which were mentioned in the research gap section. Moreover, results appear to show that, regression-based forecasting provides better efficiency compared to prediction with a classification model. However, it is only valid when up to 30 minutes gap window is shifted towards the target window for forecasting. There is a further benefit of regression-based forecasting in comparison to a binary detection or classification. The regression-based forecast generates a future estimate of the signal as a function of time. This could be a potentially rich source of clinically relevant information to, e.g. health care providers. Moreover, for the model forecasting, across a 30 minute target window in the case of a dangerous clinical event, this means that the model predicts the event right at the start of the event, rather than at the end which is the current approach to e.g. defining a hypotensive event. This allows health care professionals around 30 minutes to react even when there is no gap considered. This study shows that the forecast model with no gap window forecasts events with around 99 per cent accuracy.

1.5 Dissertation Outline

This thesis is composed of 6 chapters and is organised as follows:

- **Chapter 2 : Background**

  This chapter introduces the relevant background to this thesis. It introduces the theory of time series forecasting, forecasting strategies and data approaches. Machine learning algorithms are also proposed, along with tools in time series data analysis. This chapter also presents the background in terms of physiological data. It describes a vast source of publicly accessible physiological data known as the MIMIC database. It also describes the steps needed to process data for forecasting and event prediction purposes. Lastly, model evaluation procedures are also considered as these are used throughout the thesis.

- **Chapter 3 : Literature Review**

  This chapter includes a review of time-series data applications in the medical domain in terms of machine learning algorithms. It mainly focuses on existing techniques for time series physiological data, machine learning, forecasting and event prediction. The aim is to explore the limitations and scope of machine learning algorithms in forecasting and event prediction with physiological time series data.

- **Chapter 4: Forecasting Strategies and Data Approaches for Machine Learning of Physiological Data**

  This chapter presents a comparative study of different strategies for forecasting physiological time series data (Heart Rate and Blood Pressure). Forecast strategies are compared along with machine learning algorithms. This chapter also presents two different data approaches (univariate and multivariate data) in forecasting of physiological time series data (blood pressure) comparing univariate and multivariate machine learning models and forecasting strategies.

- **Chapter 5: Machine Learning in Intervening in Clinical Events**

  This chapter presents the scope of machine learning algorithms in the intervention of a hypotensive event and bradycardia event by analysing mean arterial blood

pressure and heart rate time series data. A novel hybrid model combining two regression algorithms are also presented in predicting hypotensive events and bradycardia events.

- **Chapter 6 : Conclusions and Future Work**

  This chapter describes the overall outcomes of the research and the contributions made as a result in intervening hypotensive event and bradycardia event after analysis physiological time series data with machine learning techniques. Limitations of the conducted research, as well as future directions, are also presented.

- The references for the thesis and a number of appendices for the thesis are also included at the end of the thesis.

Chapter 2

Background

This chapter introduces some aspects of the relevant background behind this thesis. The chapter begins by defining the concept of time series forecasting, forecasting strategies and data approaches. It also includes a description of physiological time series data. As part of this, an overview of the Medical Information Mart for Intensive Care (MIMIC) database is presented. This chapter also includes a discussion on how to prepare these data for the prediction and forecasting models implemented here in this research. An important aspect of this is on how to prepare data for a supervised machine learning algorithm using a sliding window technique. A brief introduction of machine learning along with different algorithms are presented. The chapter also describes the evaluation procedures and metrics of time series prediction.

2.1 Time Series Forecasting

Prediction of time series can be classified as time series forecasting and time series classification. Time series forecasting is considered to be an important area of signal processing and more recently machine learning. It has an ultimate goal of predicting the future values of variables of interest by exploring patterns from past data [Molaei and Keyvanpour, 2015]. Time series data can be described as

$$X = (x_t; t = 1, \ldots ...., N) \tag{2.1}$$

where X is the time series, t is time, where there are N observations during that time and x_t is the value measured at time instant t.

Shumway & Stoffer [Shumway and Stoffer, 2017] have been quoted as defining time series as "*a collection of random variables indexed according to the order they are obtained in time*". Times series forecasting has been found effective for many applications. Time series forecasting has been given much attention in the finance sector. Datasets from different areas of finance have been investigated for forecasting [Khashei et al., 2012, Li and Hu, 2012, Gupta and Wang, 2010, Zhu and Wang, 2010]. Another area of popular interest is electricity load forecasting. To forecast electricity load in advance researchers have used time series forecasting [Deng and Jirutitijaroen, 2010, Mandal et al., 2007, Ren et al., 2016]. Radzuan et al. [Radzuan et al., 2013] and Olailya & Adeyemo [Olaiya and Adeyemo, 2012] also found time series forecasting useful for weather prediction.

In general time series forecasting can be classified as short term where a single-step forecast is required, and long-term where a multi-step forecast is required. Moreover, single-step forecasting can be used for long-term time series forecasting with a long forecast horizon. Selection of forecasting class and strategy depends on the requirement of the application field and frequency of collected data.

2.1.1 Single Step Forecasting

Single step forecasting is applicable where short-term forecasting is required. For example durations of several minutes, hours or days could all be considered short-term forecasts depending on the nature of the forecasting problem. For short-term forecasting, computing a one step ahead forecast is significant. One step ahead forecast $(t+1)$ is achieved by passing the current and past observations $(t, t-1, ..., t-n)$ to a chosen model,

$$F(t+1) = M(o(t), o(t-1), o(t-2), ..., o(t-n)) \tag{2.2}$$

where $F(t+1)$ is the forecast for time $(t+1)$, M is the model and $o(t)$ is an observation at time t. In this chapter, a Multi-Input Multi-Output (MIMO) strategy is used to represent single-step forecasting.

2.1.2 Multi-step Forecasting

Multi-step Forecasting is where models require more than one step to forecast the target window. In general, multi-step forecasting is considered a challenging task [Masum et al., 2018, Masum et al., 2017]. Multi-step forecasting is useful where the field of application requires long-term duration forecasting. In this chapter, Recursive strategy, Direct strategy, DirRec strategy and DIRMO strategy all represent multi-step forecasting. Multiple steps can be used to forecast the target window as follows:

$$(F(t+1), F(t+2)) = M_1(o(t), o(t-1), o(t-2), ..., (t-n)) \tag{2.3}$$

$$(F(t+3), F(t+4)) = M_2(o(t), o(t-1), o(t-2), ..., o(t-n)) \tag{2.4}$$

where $F(t+n)$ is the forecast for time $(t+n)$, M_1 and M_2 are the models and $o(t)$ to $o(t-n)$ are observation at time t to $t-n$.

2.2 Forecasting Strategies for Physiological Data

Ben Taieb et al. describe five forecasting strategies which are all considered in the experiment here along with a forecasting strategy called RECMO Strategy [Taieb et al., 2012]. Among these strategies, Multi-Input Multi-Output (MIMO) strategy produces multiple outputs applying single-step forecasts whereas the rest of the strategies require a multiple steps to forecast multiple outputs.

2.2.1 The Recursive Strategy

In Recursive forecasting, a model is developed to forecast one step ahead, and is repeatedly used to forecast multiple outputs. In such a case, the forecast of the prior time step is fed back along with past available data to the model as input and used for forecasting the next forecast time point. For example, to forecast the next two points of any scenario using a recursive strategy, the first forecast point $F(t+1)$ needs to be calculated through a model. Then $F(t+1)$ is used as input to forecast the second

point $F(t+2)$. This is illustrated with the following:

$$F(t+1) = M(o(t), o(t-1), o(t-2), ..., o(t-n)) \tag{2.5}$$

$$F(t+2) = M(F(t+1), o(t), o(t-1), ..., o(t-n)) \tag{2.6}$$

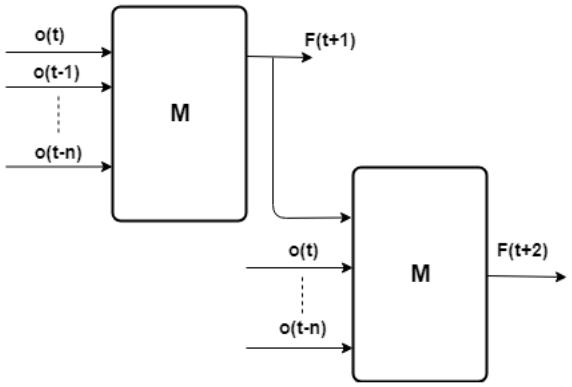


Figure 2.1: Architecture of the Recursive strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents models

Fig. 2.1 shows the architecture of the Recursive strategy for multi-step ahead forecasting. The Recursive strategy is quite simple to apply but results in high variance because of the accumulation of errors in the forecasts. Performance degrades in long-term forecasting when this strategy is used.

2.2.2 The Direct Strategy

Direct strategy develops H separate forecasting models to forecast H steps. For example, to forecast the next two points of any scenario using a Direct strategy, the first forecast point $F(t+1)$ needs to be calculated through a model and then a different model would be used to forecast the second point $F(t+2)$. The second point is therefore not dependent on the first point estimate. This example can be seen below.

$$F(t+1) = M_1(o(t-1), o(t-2), ..., o(t-n)) \tag{2.7}$$

$$F(t+2) = M_2(o(t-1), o(t-2), ..., o(t-n)). \tag{2.8}$$

The direct multi-step strategy can be expressed as

$$y_{t+h} = f_h(y_t, ..., y_{t-n+1}) \tag{2.9}$$

where h is the number of steps to forecast into the future, n is the autoregressive order of the model, f_h is any arbitrary learner.

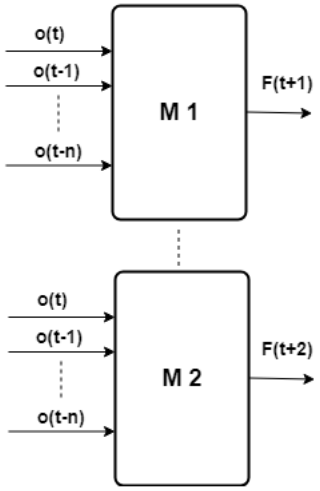


Figure 2.2: Architecture of the Direct strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents models

Fig. 2.2 shows the architecture of the direct strategy for multi-step ahead forecasting. The direct strategy does not consist of any accumulated errors because this strategy does not use any forecasting value as an input. However, this strategy does not guarantee any statistical dependence between forecasted point as every model is trained independently. This strategy requires large computational time.

2.2.3 The DirRec Strategy

The DirRec strategy is a combination of Direct and Recursive strategy which overcomes the limitations of the Recursive and Direct strategies. The DirRec strategy creates H separate models to predict H time steps. However, each model uses the predictions made by models at previous time steps as an input. Fig. 2.3 shows the architecture of the DirRec strategy for multi-step ahead forecasting. For example, to forecast the next

two points of any scenario using a DirRec strategy would require building one model M_1 to forecast the first point $F(t+1)$ and a second model M_2 to forecast the second point $F(t+2)$. The second model M_2 will use the predicted first point $F(t+1)$ of model M_1 as an input. This is illustrated with the following:

$$F(t+1) = M_1(o(t-1), o(t-2), ..., o(t-n)) \tag{2.10}$$

$$F(t+2) = M_2(F(t+1), o(t-1), ..., o(t-n)) \tag{2.11}$$

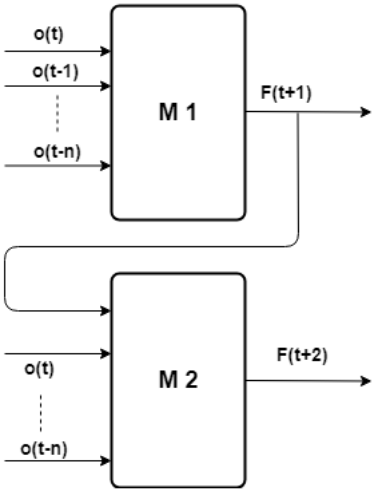


Figure 2.3: Architecture of the DirRec strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents Models

2.2.4 The MIMO Strategy

MIMO strategy only consists of one model which takes multiple inputs and forecasts multiple outputs . For example, to forecast the next two points of any scenario using a MIMO strategy would require building a single model to forecast the first point $F(t+1)$ and the second point $F(t+2)$.This is illustrated with the following:

$$(F(t+1), F(t+2)) = M(o(t-1), o(t-2), ..., (t-n)) \tag{2.12}$$

Fig. 2.4 shows the architecture of the MIMO strategy for multi-step ahead forecasting. The MIMO strategy can avoid both accumulation error and conditional independence assumption issues. However, the MIMO strategy is more complicated compared

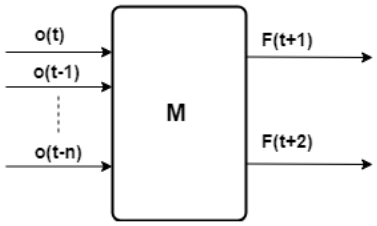


Figure 2.4: Architecture of the MIMO strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents Models

to other forecasting strategies, as it learns the dependence structure between inputs and outputs as well as between outputs. Models consisting of such a strategy are slower to train and could include overfitting issues if enough data is not available.

2.2.5 The DIRMO Strategy

DIRMO strategy is a combination of Direct and MIMO strategy. In the DIRMO strategy forecast points are partitioned in the block(s). Then, the MIMO strategy is used to forecast the values inside the blocks, and the Direct strategy continues until a forecast horizon(H) is foretasted. Fig. 2.5 shows the architecture of the DIRMO strategy for multi-step ahead forecasting.

For example, to forecast the next four points of any scenario using a DIRMO strategy would require 2 blocks to partition 4 points, which leaves 2 forecast points in each block. Now, two MIMO and Direct strategy combination will be required to forecast 4 points, where the first combination will predict the first two forecast points and the second combination will forecast next 2 forecast points. In the DIRMO strategy one model's output does not depend on another model. This is illustrated with the following:

$$(F(t+1), F(t+2)) = M_1(o(t-1), o(t-2), ..., (t-n)) \tag{2.13}$$

$$(F(t+3), F(t+4)) = M_2(o(t-1), o(t-2), ..., o(t-n)) \tag{2.14}$$

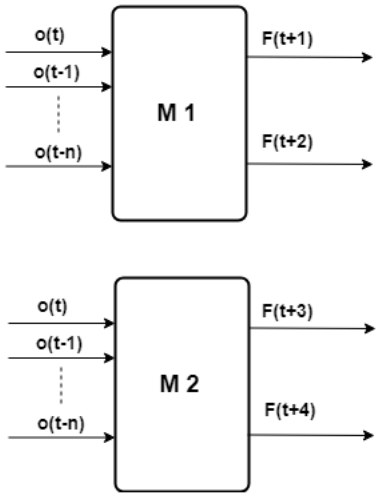


Figure 2.5: Architecture of the DIRMO strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents Models

2.2.6 The RECMO Strategy

The RECMO strategy is developed using a combination of Recursive and MIMO strategies. Fig. 2.6 shows the architecture of RECMO strategy for multi-step ahead forecasting. Forecast points are partitioned in the blocks. Then, the MIMO strategy is used to forecast the values inside the blocks, and the Recursive strategy continues until a forecast horizon (H) is foretasted. For example, to forecast the next four points of any scenario using a RECMO strategy would require two-blocks to partition 4 points. Which leaves two forecast points in each block. Now, two MIMO and recursive strategy combination will be required to forecast 4 points. Where the first combination will predict the first two forecast points, and the second combination will forecast the next two forecast points. In the RECMO strategy, first model output is fed into the second model as an input during the process of forecasting. This is illustrated with the following:

$$(F(t+1), F(t+2)) = M(o(t-1), o(t-2), ..., (t-n)) \tag{2.15}$$

$$(F(t+3), F(t+4)) = M(F(t+1), F(t+2), o(t-1), ..., o(t-n)). \tag{2.16}$$

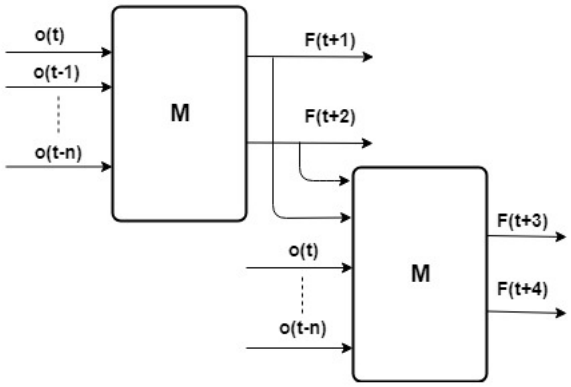


Figure 2.6: Architecture of the RECMO strategy where o(t)...o(t-n) is the observation, F(t+1)....F(t+n) is the forecast point and M represents Models

2.2.7 Comparison of Forecasting Strategies

Every strategy has advantages and disadvantages, and it is crucial to consider them in forecast model building for specific problems. Table 2.1 shows a comparison of different forecast strategies explaining the advantages and disadvantages of each strategy. Table 2.2 describes how each strategy is different considering models, model types, output size, and whether the model considers the previous forecast step during forecasting.

2.3 Machine Learning in Analysing Time Series Data

Machine learning (ML) aims at discovering patterns from data without explicit programming. ML algorithms are used to model and learn important properties from data, including the stochastic dependency between a set of input and output variables. For forecasting this can be a set of previously observed data. During the model development process, a ML algorithm is trained using a training dataset. The model is then tested on a new set of input data. For forecasting of time series data, these are introduced to the ML model to make predictions. Importantly, on the basis of experience on training data, ML algorithms provide a means to learn directly from data through computational methods rather predetermined equations or conditions. The ML algorithms learn better with a greater number of samples. The predictions are then evaluated with

Table 2.1: Comparison of forecast strategies in terms of their advantages and disadvantages.

Strategy	Advantages	Disadvantages
Recursive	1) Ideal for noise free data 2) Requires very low computational time	1) High accumulation error at the output
Direct	1) Overcomes accumulation error problem	1) Conditional independence assumption 2) Requires high computational time
DirRec	1) Helps to overcome the disadvantages of both Direct and Recursive strategy.	1) Input set grows linearly with H 2) Requires very high computational time
MIMO	1) Overcome conditional independence assumption problem 2) Requires low computational time	1) Less flexibility as single model structure, used to forecast all the horizons
DIRMO	1) Helps to balance total dependence and total independence of forecasts 2) Computational time required comparable to Direct and DirRec	1) Requires an estimation of one additional parameter
RECMO	1) Trade-off between Recursive and MIMO 2) Requires low computational time	1) Accumulation error at the output

Table 2.2: How strategies are different in terms of number of models, model type, output size and uses of previous forecast step. Where 'H' is the forecast horizon and 's' is the forecast point in each block. They are briefly described in Subsection 2.2.5 and 2.2.6.

Strategy	Number of Models	Model Type	Output Size	Iterate Previous Forecast ?
Recursive	1	Single output	1	Yes
Direct	H	Single output	1	No
DirRec	H	Single output	1	Yes
MIMO	1	Multiple output	H	No
DIRMO	H/s	Multiple output	s	No
RECMO	H/s	Multiple output	s	Yes

different evaluation metrics. If the metrics are considered acceptable then the model is ready to be deployed in the field of applications. Arthur Samuel [Samuel, 1988] defined machine learning as "*The field of study that gives computers the ability to learn without*

being explicitly programmed". Tom Mitchel [Mitchell, 1997] defined machine learning as "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*". It is important to understand the context of ML in comparison to the more general discipline of Artificial Intelligence (AI).

In general, ML techniques can be applied to a multitude of problems. The ever-growing number of available computer based medical time series data are also ripe for AI developments. This can potentially include the forecasting of patient health conditions during critical care e.g. hypotensive events. Clifton et al. state that machine learning techniques will be used to solve the latest medical problems in cooperation with medical practitioners [Clifton et al., 2015]. Researchers found evidence that decision making in the medical treatment area is more suited for modelling rather than intuition alone [Schaefer et al., 2005]. Features can be extracted from large amounts of data to provide computationally convenient and relatively more efficient representations. These features are usually automatically extracted.

2.4 Supervised Learning

Machine learning can be mainly divided in to three types of techniques or algorithms [Bonaccorso, 2017] [Gollapudi, 2016]. The categories are supervised learning, unsupervised learning and reinforcement learning. A supervised learning algorithm helps a model to learn the dependencies and relationships between the inputs and a set of outputs. This helps the model to predict the target for previously unseen features or inputs. Supervised learning can be categorised into two techniques or algorithms. These are known as classification and regression [Gollapudi, 2016]. Classification algorithms are used to predict a discrete response to a problem. For example, whether an email is spam or genuine. Regression algorithms are used to predict response of a system over a period of time or some other variable. For example, how the temperature changes for the upcoming week.

For example in Fig. 2.7 addition of two numbers 5 and 6 has the result 11. The inputs are 5 and 6 and then the target is 11. This is used to train the model by which a model can be trained to 'learn' the underlying relations between the inputs and the

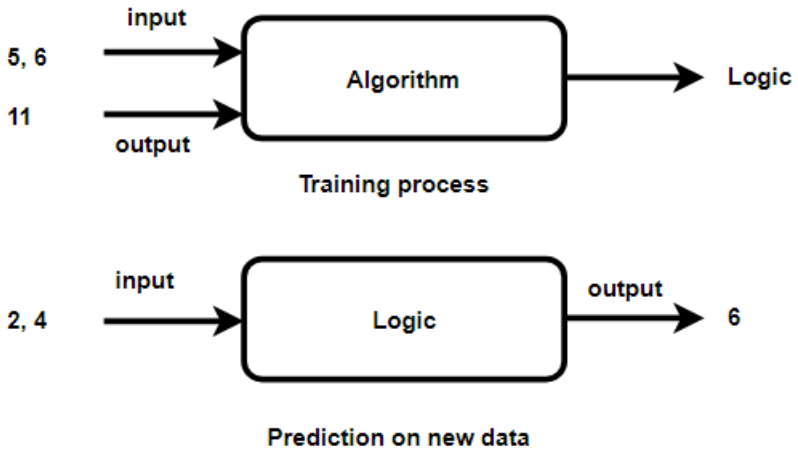


Figure 2.7: Supervised machine learning

resulting target. Then a trained model can take new data and predict the output. Supervised learning problems can be classified into two groups, known as:

- **Classification problems:** where the output variable is a category. For example, 0 and 1. Classification algorithms are used to solve such problem.
- **Regression problems:** where the output variable is a real value. For instance, a weight. Regression algorithms are used to solve such problem.

2.4.1 Difference Between Classification and Regression Algorithms

Classification and regression algorithms are both supervised machine learning algorithms. The most important aim of both algorithms is to learn a model based on training data through a process referred to as supervised training. It is also very important to know the difference between classification and regression. The main difference can be seen from the output. The output variable in classification is discrete and the output variable in regression is continuous. Moreover, it can be stated that classification predicts a label whereas regression forecasts a quantity. Predictive modelling is a problem where a model is developed which will learn from historical labelled data and make a prediction on unlabelled data. Such a model learns an approxima-

tion function $Y = f(X)$; where, f is the mapping function, X is a vector or sequence learning of input variables and Y corresponds to the output variables. The aim of the algorithm is to explore the best mapping function from available resources.

2.5 Data and Processing

This section discusses physiological time series data and their sources. As part of this, an overview of the Medical Information Mart for Intensive Care (MIMIC) database is presented. This section also discusses how to prepare data for the prediction and forecasting models implemented here in this research. It also shows how to prepare data for a supervised machine learning algorithm using a sliding window technique.

2.5.1 MIMIC Database

The MIMIC II and III databases are freely accessible critical care databases containing electronic health records data of the Intensive Care Units (ICU)s at the Beth Israel Deaconess Medical Center in Boston, Massachusetts [Johnson et al., 2016b,Saeed et al., 2011]. Advantages of using the MIMIC databases is that all data are anonymised and open to researchers. A basic overview of the MIMIC database is shown in Fig. 2.8

Three types of data are available on the MIMIC database; these are clinical data, physiological data and mortality data. Clinical data consists of information about hospital administration, physiologic, medications, lab tests, notes and reports. Physiological data consists of information of a multi-parameter physiological signal in the form of wave and time series are obtained from ICU patient bedsides. The multi-parameter physiological signals include blood pressures, heart rate and oxygen saturation. The waveform data were sampled at 125 Hz but are also made available at a lower sampling frequency. The signals used as part of the work here, for the time-series forecasting and prediction, are sampled every minute. This is in common with a number of the existing literature [Ghosh et al., 2016, Lee and Mark, 2010b, Chen et al., 2009, Rocha et al., 2011, Langley et al., 2009, Ghaffari et al., 2010].

The MIMIC databases provide open source access to researchers to highly granular ICU data, and is of critical importance e.g. for machine learning experts, to help make contributions to the field of critical care decision making. The MIMIC waveform

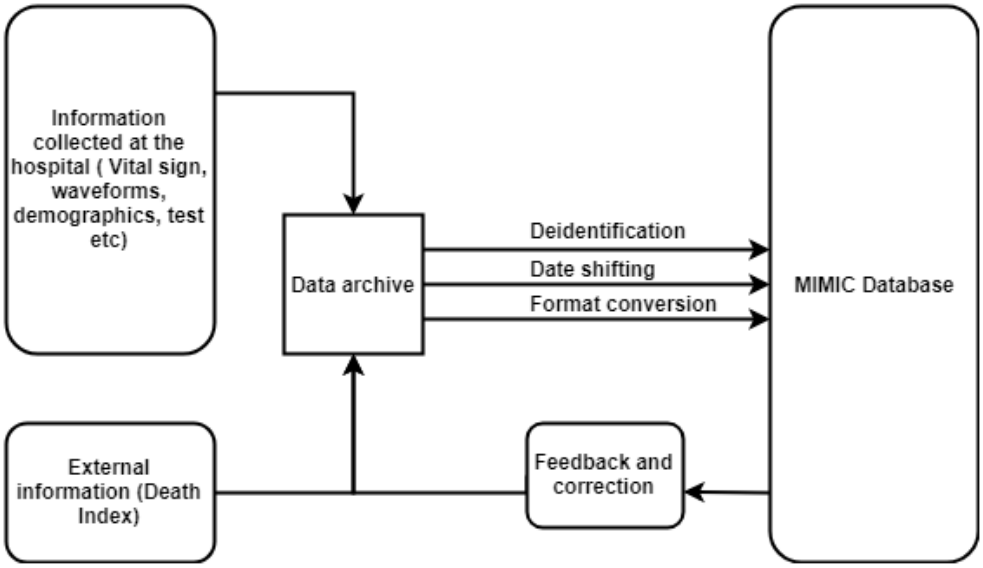


Figure 2.8: Medical Information Mart for Intensive Care (MIMIC) database overview of the data collection and archiving process.

databases are available on the PhysioNet website. A user can download it without any registration process. However, to access the clinical information in addition to the waveforms, a user is required to complete a simple data use agreement and human subjects training. Extracting and processing data from the MIMIC database is a very complex task. However, the MIMIC website provides detailed documentation and procedures to help the user obtain the required data from the database. The MIMIC-II and MIMIC-III databases currently contain information on approximately 36,000 and 53,423 patients' respectively. More than 1700 researchers worldwide in academia, industry and medicine have accessed the MIMIC database. Their research includes physiological signal processing, clinical decision support, predictive algorithms in critical care, pharmacovigilance, natural language processing and many more.

The work in this thesis is related to the topic of physiological time series modelling and data processing. Minute-by-minute physiological time series data (blood pressure and heart rate) were extracted from the matched subset of the MIMIC database. As an example, a number of different physiological time series data for a single patient are presented in Fig. 2.9.

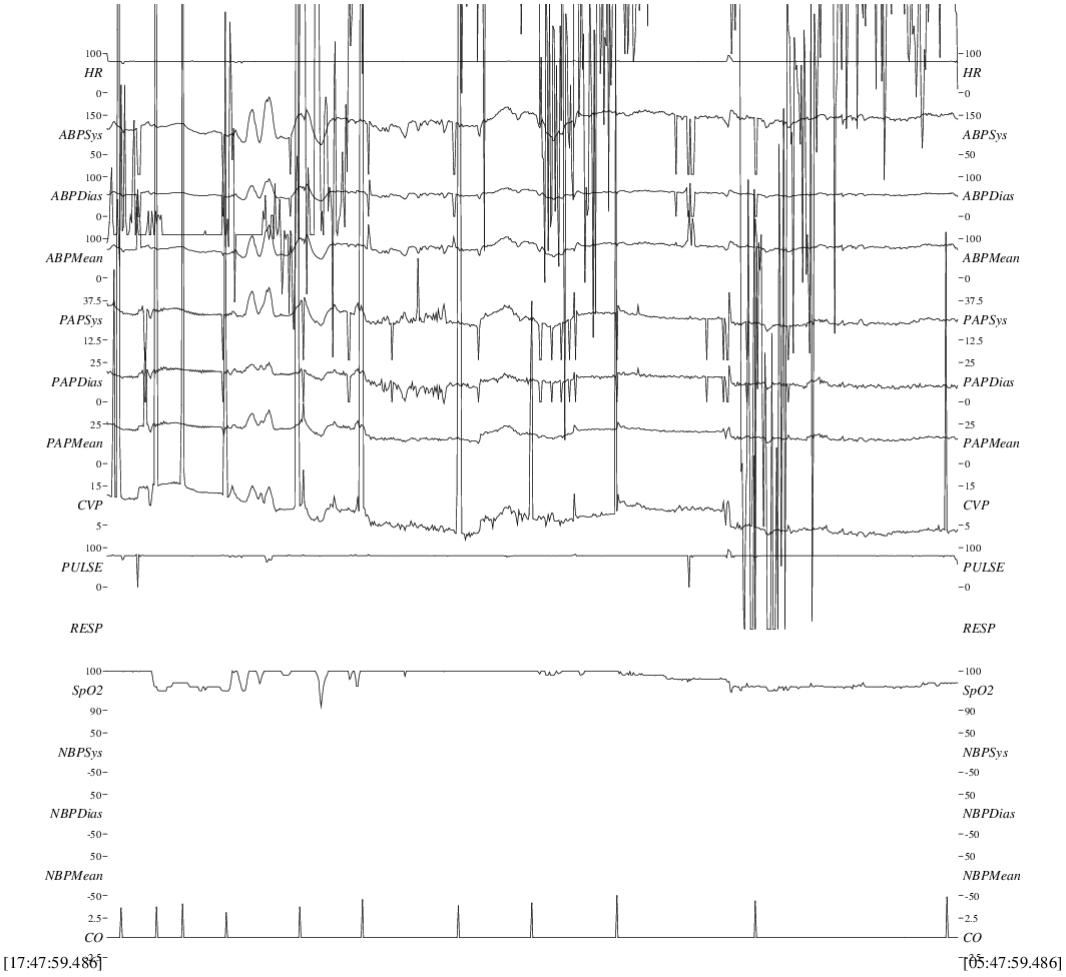


Figure 2.9: Sample data for a single patient stay in a medical intensive care unit

2.5.1.1 Blood Pressure and Heart Rate

Blood pressure describes the pressure that is exerted on the arterial wall by blood. Peak pressure on the arterial wall is called systolic blood pressure and the lowest pressure on the arterial wall is called diastolic blood pressure [Klabunde, 2011]. Measurement of blood pressure provides the medical professional with an early indication of many diseases. It therefore helps them to treat individuals for these diseases. Traditionally, blood pressure is measured with a sphygmomanometer. More advanced blood pressure monitors are available on the market. Some are able to measure blood pressure from the wrist. These might show the output digitally on a display unit and they may

have the ability to transmit the results wirelessly. There is a generally accepted Blood Pressure (BP) classification, which is provided in Table 2.3 [Chobanian et al., 2003], which show that high blood pressure leads to hypertension and low blood pressure leads to hypotension.

Table 2.3: Blood pressure classification

BP Classification	**Systolic BP, mm Hg**	**Condition**	**Diastolic BP, mm Hg**
Normal	<120	and	<80
Prehypertension	120-139	or	80-89
Stage 1 hypertension	140-159	or	90-99
Stage 2 hypertension	$\geq$ 160	or	$\geq$ 100
Hypotension	<90	or	<60

Heart rate is the number of beats that an individual's heart makes in 60 seconds. Heart rate is measured in beats per minute (BPM). Heart rate is the electrical activity of the heart and function of time.

Table 2.4: Classification of resting heart rate

Classification	**Resting heart rate**
Bradycardia	<60 bpm
Normal	60 - 100 bpm
Tachycardia	>100 bpm

A common classification of resting heart rate is presented in Table 2.4. The standard resting heart rate (HR) of a healthy adult is 60-100 beats per minute. The range below can be classified as bradycardia and the range above is classified as tachycardia. In both of these states, an ECG is essential for further investigation [Klabunde, 2011].

2.5.1.2 Univariate Time Series Data

In univariate time series, a single variable is recorded sequentially over equal time increments. For example, Table 2.5 consists of the time-dependent (each minute) Heart Rate (HR) values of a patient. Here, only a single variable (HR) is recorded. A

Table 2.5: Univariate time series data

Time and date	HR
31/07/2176 18:49	107.9
31/07/2176 18:50	108.8
31/07/2176 18:51	109.6
31/07/2176 18:52	110.5
31/07/2176 18:53	111.1
31/07/2176 18:54	113.1
31/07/2176 18:55	112.5
31/07/2176 18:56	112.1
31/07/2176 18:57	112.2
31/07/2176 18:58	112.6
31/07/2176 18:59	112.6
31/07/2176 19:00	114.1
31/07/2176 19:01	114.2
31/07/2176 19:02	111.3
31/07/2176 19:03	109.5
31/07/2176 19:04	107.9
31/07/2176 19:05	106.6
31/07/2176 19:06	106
31/07/2176 19:07	105.7
31/07/2176 19:08	104.9

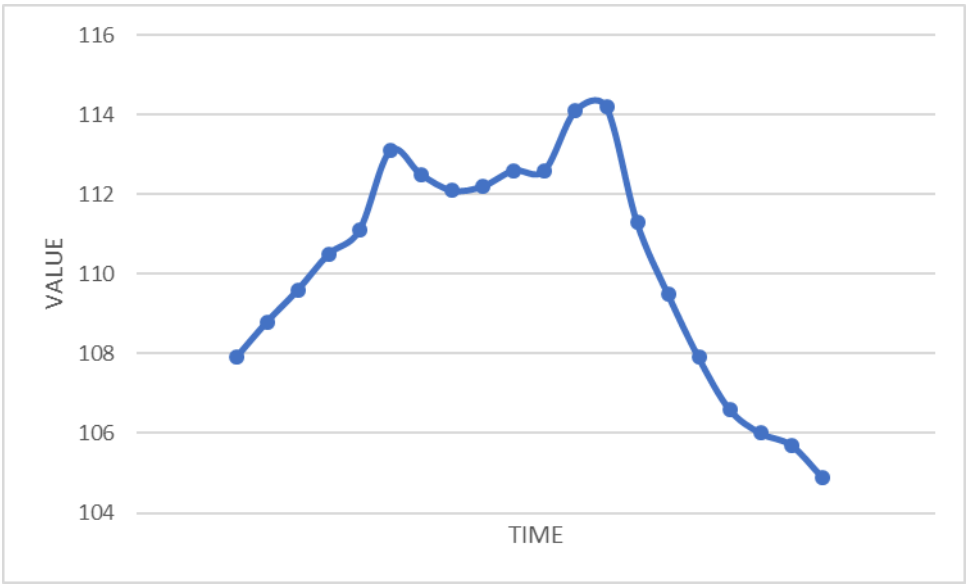


Figure 2.10: Univariate data

forecasting model can predict the future values of HR using the univariate data given in Table 2.5. Such forecasting is termed as univariate time series forecasting. A plot of the univariate data is shown in Figs. 2.10.

2.5.1.3 Multivariate Time Series Data

In Multivariate time series, more than one variable is recorded sequentially over equal time increments. For example, Table 2.6 consists of the time-dependent (each minute) HR, BP and respiratory rate values of a patient. Here three variables are recorded, and such type of time series data is called multivariate time series data. A forecast model can predict the future values of HR using the multivariate data given in Table 2.6. Such forecasting can be stated as multivariate Time Series Forecasting. Plots of the multivariate data is shown in Figs. 2.11. In multivariate data, each variable has a dependency on others variable and such dependence could be useful for forecasting future values of a specific variable. However, it is hard to find and extract such data in real-life cases as in most cases data are collected in different frequency.

Table 2.6: Multivariate time series data

Time and date	HR	ABP	RESP
31/07/2176 18:49	107.9	75.6	21.3
31/07/2176 18:50	108.8	80.8	23.7
31/07/2176 18:51	109.6	90.8	27.1
31/07/2176 18:52	110.5	93.1	22.6
31/07/2176 18:53	111.1	94.3	23
31/07/2176 18:54	113.1	98.3	24.5
31/07/2176 18:55	112.5	95	24.6
31/07/2176 18:56	112.1	92.9	22.2
31/07/2176 18:57	112.2	92.3	22.5
31/07/2176 18:58	112.6	89.2	23.3
31/07/2176 18:59	112.6	88.4	21.7
31/07/2176 19:00	114.1	93.5	24.8
31/07/2176 19:01	114.2	85.7	24
31/07/2176 19:02	111.3	75.1	21.6
31/07/2176 19:03	109.5	69.5	22.2
31/07/2176 19:04	107.9	66.6	21.2
31/07/2176 19:05	106.6	63.4	21.6
31/07/2176 19:06	106	65.2	22.5
31/07/2176 19:07	105.7	65.7	22.9
31/07/2176 19:08	104.9	63.1	22.8

Figure 2.11: Multivariate data

2.5.2 Extracting Knowledge from The Time Series Data

Forecasting can be performed using a number of different techniques ranging from traditional statistical-based regression, signal processing through to machine learning and even deep learning methods. In terms of machine learning, time series can be forecasted using several machine learning techniques. It is essential to reframe the time series when using a machine learning technique. Time series data needs a representation of pairs of input and output sequences from a sequence. This reframing or representation of data will enable machine learning algorithms to be trained on the time series forecasting problem. Supervised learning is a widespread phenomenon in practical machine learning. In supervised learning, a machine learning algorithm is used to learn a mapping function f between the input variables X and output variables Y,

$$Y = f(X). \tag{2.17}$$

The aim is to teach the model well during the training process so that for a new input data X', the model can predict the output variables Y' for that data.

2.5.2.1 Data for Supervised Prediction

For a classification problem, time-series data along with temporal dependencies can be used for prediction purposes. Table 2.7 shows a sequence of specific length, maintaining a temporal dependence which is chosen and associated with a class. For example, in Table 2.7, sequence (2,4,6,8,10) is a time-series sequence with a temporal dependency with class 0 associated with it. Class 1 is associated with the sequence (1,3,5,7,9). Each of these sequences are called samples by which the model is trained. The prediction model will accept the same length sequence as an input to predict the class.

Table 2.7: Example sequences and their associated classes for a classification problem with supervised learning.

Sequence	Class
2, 4, 6, 8, 10	0
1, 3, 5, 7, 9	1

2.5.2.2 Data for Supervised Forecasting

For forecasting time series data needs to be restructured to make it a supervised learning problem . This can be done by taking preceding time steps as input variables and the next time step as the output variable. For example, the Table 2.8 is an example of a univariate time series data.

Table 2.8: Univariate time series data.

time	value
1	10
2	11
3	13
4	15
5	12

This univariate time series dataset in Table 2.8 can be rephrased as a supervised learning problem. This is done by applying preceding time steps as input variables

(X) and the next time step as the output variable (Y). Following this, the data would appear as presented in Table 2.9.

Table 2.9: Supervised learning of univariate time series data

X	Y
?	10
10	11
11	13
13	15
15	12
12	?

Some observations are provided below after comparing the transformed dataset with the original dataset. These are:

- The previous time step is the input (X), and the next time step is the output (Y) in a supervised learning problem.
- The order between the observations is preserved.
- There is no previous value that can be used to predict the first value in the sequence. There is also no known next value to predict for the last value in the sequence. During the training process, these rows will be deleted.

The method used to transform the dataset is known as the sliding window method or lag method [Dietterich, 2002]. The number of preceding time steps used as predictions is called the window width or size of the lag. For the example above, the window width is one as a single variable or feature are used as the input (X) and a single variable is used at the output(Y). This technique can also be referred to as a univariate approach.

2.5.2.3 Sliding Window Method with Multivariate Time Series Data

Table 2.10 shows a multivariable time series dataset where two observations are available at each time step. Data will be transformed into a supervised problem in predicting feature 2.

Table 2.10: Example of multivariate time series data for supervised learning.

Time	Feature 1	Feature 2
1	2	80
2	4	85
3	7	87
4	9	89
5	5	82

The exemplar time series dataset will be reframed, again considering a window width of one. So, the supervised dataset will have two previous time steps of feature 1 and one previous time steps of feature 2. These will then be used to predict the next time step value of feature 2. Supervised datasets will consist of three input features and one output feature in each sample, as presented in Table 2.11. In the table first and the last row will be deleted during the training process.

Table 2.11: Supervised learning of single-step prediction for the exemplar multivariate data in Table 2.10.

X1	X2	X3	Y
?	?	2	80
2	80	4	85
4	85	7	87
7	87	9	89
9	89	5	82
5	82	?	?

The above scenario, where multiple variables or features are used as input (X) and a single variable as an output (Y), can be defined as a multivariate approach.

In all the above cases, just single-step prediction has been described. Data has been reframed for prediction of the single output variable. But it could be required to predict multiple steps simultaneously. This is known as multi-horizon forecasting.

2.5.2.4 Sliding Window Method with Multi-Step Forecasting

The number of time steps ahead to be forecasted is important. Multi-step ahead forecasting is where two or more future time steps are forecasted. Sliding window methods for multi-step ahead forecasting will be discussed here. Multi-step ahead forecasting will be illustrated with the exemplar univariate time series data as shown in Table 2.8. First it will be rephrased as a supervised learning problem through the sliding window method. Following this, the dataset would appear as shown in Table 2.12.

Table 2.12: Supervised learning of multi-step prediction

X1	Y1	Y2
?	10	11
10	11	13
11	13	15
13	15	12
15	12	?
12	?	?

Here, the selected data is reshaped as a two-step forecasting dataset for supervised learning, again with a window width of one. The first and last two rows cannot be used for training of a supervised model. In the above example, only one window width is used to predict the next two steps of the problem. This can obviously be a burden for real-life applications. Appropriate experimentation is required to find a suitable window width for any real-life application.

2.5.3 Data Transformation

Transformation of time series data is often required when building prediction and forecasting models with machine learning algorithms. There are many transformation techniques out there such as differencing, standardisation and normalisation. Differencing is used to remove the trend and seasonal structure. Different algorithms require data to be in different forms for appropriate modelling. For example, neural networks require data to be normalised prior to modelling. Standardisation and normalisation

techniques help to scale the data. The scaling process helps to change the range of values which the data covers without changing the shape of the data. Moreover, the original data scale and or shape can be recovered at the output of any model through inverse transformations. Different data transformation techniques will be used to convert the original data forms which are acceptable by machine learning algorithms.

2.5.3.1 Differencing

A systematic structure such as trend or seasonality on time series data can easily be removed by differencing. For example, first-order differencing is applied to time series data to remove the trend from the data. First-order differencing involves subtracting the previous value from each value in the time series data as shown in (2.18). Then, the inverse of the difference is applied to extract the original value, as shown in (2.19). This process can be repeated N times.

$$\text{difference}(t) = \text{observation}(t) - \text{observation}(t-1) \tag{2.18}$$

$$\text{inverted}(t) = \text{differenced}(t) + \text{observation}(t-1) \tag{2.19}$$

2.5.3.2 Standardization

Standardisation transforms data in the form of a standard normal distribution. The standardisation process subtracts the mean and divides the result by the standard deviation of the data sample. This transforms the data to have a mean of zero with a standard deviation of 1. The procedure of standardisation can be expressed by Equation 2.20. To standardise the features, the sample mean vector $\boldsymbol{\mu}$ is subtracted from every training sample and then every training sample is divided by the standard deviation σ,

$$\mathbf{x}_{\text{std}} = \frac{\mathbf{x} - \boldsymbol{\mu}}{\sigma} \tag{2.20}$$

where $\mathbf{x}$ is the vector consisting of the n^{th} feature values of all training samples.

2.5.3.3 Normalization

The normalization process rescales the original data to have the range of [0 - 1]. Normalization can rescale the variables to be constrained to a specific range through min-max scaling. To normalize the data, the min-max scaling is applied to the i^{th} variable of each feature vector. The new value of a sample can be calculated as follows:

$$z_i = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \tag{2.21}$$

where, x_i is the i^{th} variable in a feature vector, $\min(\mathbf{x})$ is the smallest value for that variable in the feature vector and $\max(\mathbf{x})$ is the largest value.

2.6 Classification Algorithms

Classification aims to assign a class to an unlabelled set of data from the experience of training data. Classification is the most common type of problems in machine learning. It can be used to help solve common problems in different applications. Classification can be further categorised based on the type of output variable into binary class, where output is binary and multiclass, where the categorical output variable has multiple (>2) categories. An example of a binary classification problem is email classification problem where an email can be classified as spam (0) or not spam (1). An example of multiclass classification is news categorization. For instance, there are different categories of news, such as politics, international, sports, economy.

2.6.1 Logistic Regression

Logistic regression (LR) is a classification algorithm used for binary classification [Murphy, 2012]. Despite its name, it should be noted that it is not a regression algorithm. Logistic regression calculates the relationship between categorical dependent variables (output) and independent variables (input) by estimating probabilities. This can be described as:

$$p(y|\boldsymbol{x}, \boldsymbol{w}) = Ber(y|sigm(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x})), \tag{2.22}$$

where $\boldsymbol{x}$ are inputs, $\boldsymbol{w}$ are weights and y is the label or output. Here y has a Bernoulli distribution and a sigmoid function. The sigmoid function is defined as

$$\sigma(\boldsymbol{w^T x} + \boldsymbol{b}) = \frac{1}{1 + e^{-(\boldsymbol{w^T x} + \boldsymbol{b})}} \tag{2.23}$$

LR model ensures that the probabilities of all possible labels sum up to 1. Each individual probability has a value between 0 and 1. The sigmoid function maps the whole real line to [0, 1]. This is necessary for the output to be interpreted as a probability. The model's prediction y_{pred} is the class; whose probability is maximal:

$$y_{pred} = \arg\max_i \boldsymbol{P}(y = i|\boldsymbol{x}, \boldsymbol{w}) \tag{2.24}$$

LR algorithm has been used in medicine [Bagley et al., 2001] to predict heart failure [Wu et al., 2010], cardiac arrest [Panahiazar et al., 2015], re-hospitalization [Zolfaghar et al., 2013] and ICU transfer [Zhai et al., 2014].

2.6.2 K-Nearest Neighbors

In KNN, predictions can be made for a new data point ($\boldsymbol{x_0}$). A majority vote among the k neighbors which are closest in distance d_i to $\boldsymbol{x_0}$ [Bishop, 2006] while searching the training dataset. A number of distance functions are possible to calculate the neighbour. The most popular one is the Euclidean distance, given as:

$d_i = \|\mathbf{x}_i - \mathbf{x}_0\|$

The KNN is also known as a lazy learner as the learning is implemented only when a prediction is required. The limitation of KNN is that it takes a long time to classify, but this could be solved by reducing the dimensions of the data or attributes.

KNN has been used to predict heart failure [Isler, 2016, Narin et al., 2014, Masetic and Subasi, 2016], late-onset neonatal sepsis [Mani et al., 2014].

2.6.3 Decision Trees

Decision trees are amongst the most used classification algorithms in machine learning. A decision tree builds a classification model by creating a flowchart-like tree structure. The decision tree begins by placing an attribute at the root node from training data using a recursive divide-and-conquer algorithm. A new branch is created for each

value in the attributes and the process repeats recursively until the instances reach that branch [Witten et al., 2016]. A node is called an ideal node if all the instances which map to it belongs to the same class. Entropy and information gain are used to construct a decision tree. Entropy calculates the predictability of an event. Information gain measures the change in entropy with respect to the independent attribute. It also calculates the information represented by each attribute. Decision tree construction is all about looking for an attribute which returns the highest information gain. More information can be found in [Quinlan, 2014].

Decision trees have been used to predict heart failure [Son et al., 2012], and heart failure destabilizations [Pocock et al., 2005]. Overfitting is an issue for deep decision trees but random forests overcome the overfitting issue by creating a random subset of decision trees. Moreover, it takes the average of all predictions. The ability to add randomness in the model and the search for the best feature among a random subset of features can make random forests a useful model.

2.6.4 Random Forests

Random forests are considered to be one of the most powerful classification algorithms. Random forests are an ensemble machine learning algorithm. They take an average of the predictions of a large number of decision trees [Breiman, 2001]. Random forests are good at generalizing and perform much better than individual trees. They also require very little parameter tuning [Raschka, 2015]. Ensemble learning methods combine bagging and random selection of features. This is to construct a collection of decision trees with controlled variance. Given the independent variable T, the random forest learner combines K decision tree classifiers $h_1(T)$, $h_2(T)$, ... ,$h_K(T)$. Each $h_i(T)$ gives a classification, which is known as a voted classification. Then, the forest algorithm selects the classification having the most votes.

Random forests have been used for the prediction of heart failure [Masetic and Subasi, 2016,Austin et al., 2013], heart failure destabilizations [Guidi et al., 2014,Guidi et al., 2015], re-hospitalizations [Zolfaghar et al., 2013] [Vedomske et al., 2013] and cardiac arrest [Kwon et al., 2018] among others.

2.6.5 MLP

Multi-Layer Perceptron (MLP) is a model that is trained with a supervised learning algorithm. It learns a function with an input and an output corresponding to the labels with dimension o, corresponding to the training data with dimensionality m.

$$f(\cdot) : R^m \rightarrow R^o \tag{2.25}$$

MLP is a kind of artificial neural network. MLP can be applied to both classification and regression problems [Trevor et al., 2009]. MLPs are fully connected multilayer feedforward networks which are composed of different layers. The first layer of the MLP represents the input of the network whereas the last layer represents the output. Between the first and last layers there are one or more layers called hidden layers. The input layer contains a set of neurons

$$\{x_i | x_1, x_2, ..., x_n\} \tag{2.26}$$

which represent the input features. The neurons in the hidden layer take the values from the previous layer and apply a weighted linear summation

$$w_1 x_1 + w_2 x_2 + ... + w_n x_n \tag{2.27}$$

which is then followed by a non-linear activation function

$$g(\cdot) : R \rightarrow R \tag{2.28}$$

The last hidden layer passes the values to the output layer and the output layer returns the output values.

MLPs have been used to predict e.g. Heart Failure [Elfadil and Ibrahim, 2011, Jovic and Bogunovic, 2011, Narin et al., 2014] and cardiovascular disease [Oresko et al., 2010].

2.7 Regression Algorithms

There are cases where what we are interested in are not discrete classes. Instead, it might be a continuous variable. In many instances these types of problems are refereed to as regression type problems. The aim of regression analysis is to understand how

changes to the input consisting of independent variables can effect changes to some dependent continuous variable. The simplest regression problems are linear and involve fitting a straight line to a set of data in order to make a prediction. This is usually done by minimizing the sum of squared errors in each instance in the training set. Typical regression problems include estimating the likelihood of a disease given a range and severity of symptoms or predicting test scores given past performance.

2.7.1 AutoRegressive Integrated Moving Average (ARIMA)

There are many different approaches available for time series forecasting which have been described in [Khashei and Bijari, 2011]. AutoRegressive Integrated Moving Average (ARIMA) approach is considered to be the most common approach for time series forecasting. The ARIMA approach was introduced by Box and Jenkins [Box et al., 2015]. It has been found to be efficient for short term forecasting [Meyler et al., 1998]. The ARIMA acronym represents the key characteristics of the model. These can be represented as: AR(Autoregression): representing the relationship between the current observation and past observations; I(Integrated): the differencing of actual observations in order to make time series stationary; and MA(Moving Average): lags of the forecast errors of the moving average model.

An autoregressive process of order p, i.e., AR(p), can be described as:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \tag{2.29}$$

where, x_t is the actual value and w_t is the random error at time t; ϕ_i are the autocorrelation coefficients where, $i = 1, 2, .., p$. A moving average process of order q, i.e., MA(q), can be described as:

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}, \tag{2.30}$$

where, θ_j are the weights applied to the current and prior values of a stochastic term in the time series where j = 1,2,..,q and $\theta_0 = 1$.

If the process of ARMA is dynamic and non-stationary, then a transformation of the series is required to make it stationary [Box et al., 2015]. This results in the ARIMA model. This is achieved by replacing the measured values y_t with the results

of a recursive differencing process $\nabla^d y_t$. The number of times the differencing process has been applied is indicated by d. The first order differencing can be expressed as

$$\nabla^d y_t = \nabla^{d-1} y_t - \nabla^{d-1} y_{t-1}. \tag{2.31}$$

These three parts are parametrized in ARIMA models. The standard notation for the ARIMA model is usually given as ARIMA(p, d, q); where p is the number of lag observations, d is the degree of differencing and q is the size of the moving average window. In the ARIMA model, the forecast value is a linear compound of past values and past errors, expressed as follows,

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t + \theta w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q x_{t-q}, \tag{2.32}$$

where: x_t is the actual value and w_t is the random error at t; ϕ_i and θ_j are coefficients; and $t - p$ and $t - q$ are the orders of the autoregressive and moving average parts respectively.

ARIMA models is very popular for time series forecasting. However such models do not assume any prior knowledge about the underlying model. Models only depend on past data and error values and such characteristics make the model more robust and easy to explain. Moreover, models are created on hypothesis that the time series is produced from a linear process. Researchers found that the ARIMA model performs better on linear time series data and stationary data compared to nonlinear and nonstationary data [Kam, 2015,Lineesh et al., 2010]. This makes such a model not suitable for real-world problems where most of the time series data is nonlinear and nonstationary. Moreover, ICU data are found non-stationary [Clifford et al., 2008] [Billis and Bamidis, 2014] as the physiologic state of a patient differ over time during ICU stay. Machine learning techniques to adapt with very little prior assumptions and have the ability to learn from original data in order to generalise. This makes machine learning technique more suitable for non-linear problems. Moreover, the comparison between ARIMA and different machine learning techniques has been explored by many researchers in different fields. Ho et al. in 2002 compared ARIMA and RNN models for time series prediction and concluded that RNN performs better than ARIMA [Ho et al., 2002]. Fu et al. also found an RNN model performed better than ARIMA model for predicting traffic flow [Fu et al., 2016] in 2016. Cao et al. explored RNN models and found that

they outperform ARIMA models in forecasting wind speed [Cao et al., 2012]. Ma et al. compared LSTM with an ARIMA model in forecasting traffic speed prediction in 2015 and found LSTM outperforms the ARIMA model [Ma et al., 2015]. Tian et al. forecasted short-term traffic flow using LSTM and showed that LSTM performs better than most non-parameteric models [Tian and Pan, 2015].

2.7.2 Recurrent Neural Network(RNN)

A traditional Neural Network(NN) treats all inputs and outputs as being independent of each other also known as feedforward network. Such networks do not consider the order of time and hold any memory of the recent past. For time series forecasting it would be unwise to use such a network. An alternative is to use an Recurrent Neural Network (RNN) which includes consideration of the dependencies for past observations and thus has been found to be more effective for time series forecasting [Jain and Medsker, 2000]. The typical architecture of an RNN is shown in Fig. 2.12 which consists of input I which is fed into a neural network block A with an output O.

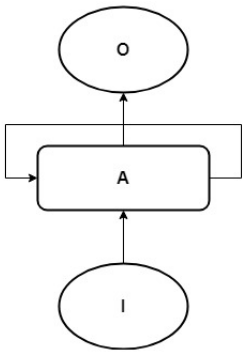


Figure 2.12: Simple architecture of a RNN where I represents input, O represents output and A represents Neural Network

A NN passes information through a loop. This looping process can be unrolled. The unrolled architecture of a full RNN network is illustrated in Fig. 2.13 for time steps 1, 2, 3 up to time t; where I_1 , I_2, I_3, ..., I_t are the inputs; A_1 , A_2, A_3,....., A_t are the hidden states or the memory of the network; and O_1 , O_2, O_3,....., O_t are the outputs. For example, A_2 is the hidden state at time step 2, where A_2 is calculated

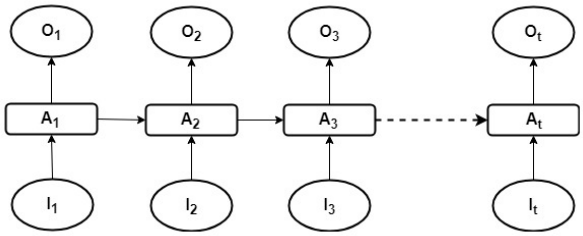


Figure 2.13: Unrolled architecture of RNN where I_n represents input, O_n represents output and A_n represents Neural Network

using a function (normally tanh) in which the function computes using a previously hidden state A_1 and the current input I_2 with,

$$A_2 = f(UI_2 + WA_1) \tag{2.33}$$

where U and W are parameters. It can therefore be concluded that an RNN does capture information that has been calculated and uses it for long term forecasting. Based on the above description of an RNN it appears to pass some potentially useful properties for long-term forecasting and perhaps it is even capable of handling long-term dependencies. In practice, however, these characteristics do not hold as shown by Bengio, et al. [Bengio et al., 1994]. Long-Short-Term-Memory (LSTM) introduced by Hochreiter & Schmidhuber [Hochreiter and Schmidhuber, 1997] in 1997 is able to learn long-term dependencies better than the RNN architecture.

The aim of an RNN is to provide an accurate outcome based on sequential data. An RNN achieves this through back propagation of the error via gradient descent. Gradient values updates the weights of a neural network with regard to the change in error. Gradient values fine tune the weights to decrease error. However, the error signal exponentially decreases during backpropagation while training. This is because the layers and time steps of an RNN relate to each other through multiplication and the derivatives are liable to the vanishing gradient problem. Due to the vanishing gradient issue, the layers closer to the input do not get trained. More importantly, RNN fails to track long term dependencies.

In an unrolled RNN, the hidden vector h_t and the output y_t are computed with:

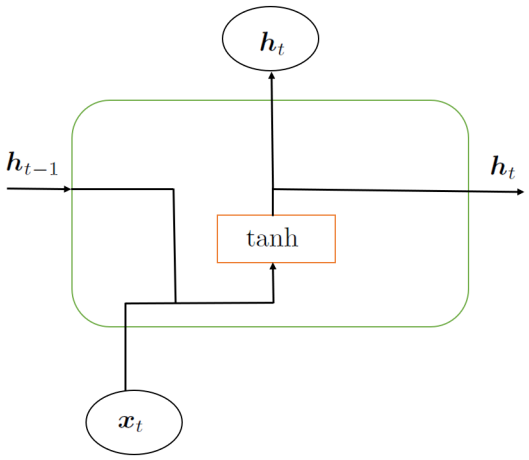


Figure 2.14: RNN architecture where x_t is the input and h_t is the output

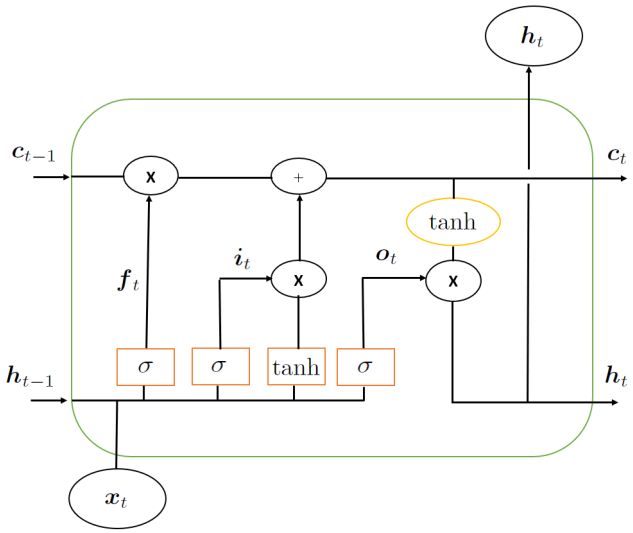


Figure 2.15: LSTM based RNN architecture where where x_t is the input, h_t is the output, c_t is the cell state, f_t is the forget gate, i_t is the input gate, o_t is the output gate, σ is the logistic sigmoid function

$$h_t = tanh(W_i x_t + W_c h_{t-1}) \tag{2.34}$$

$$y_t = W_o h_t \tag{2.35}$$

where, W_c,W_i,W_o are the set of weights of cell, input and output; h_t is the hidden vector at time t; and $tanh$ is the hyperbolic tangent function which is used as an activation

function.

The gradient of E is computed with respect to the cell state weights W_c while training the RNN through back propagation. The overall error gradient $\frac{\partial E_t}{\partial W_c}$ is equal to the sum of the error gradients at each time step [Pascanu et al., 2013b]. For step t, the multivariate chain rule is used to derive the error gradient as:

$$\frac{\partial E_t}{\partial W_c} = \sum_{i=0}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_i} \frac{\partial h_i}{\partial W_c} \tag{2.36}$$

where, E_t is the error at time t; E_t is a function of output y_t; y_t is the output at time t; h_t is the hidden vector at time t. E_t can be computed with:

$$E_t = \sum \frac{1}{2}(target - output)^2 \tag{2.37}$$

The partial derivative term of hidden state $\frac{\partial h_t}{\partial h_i}$ in (2.36) is computed through another chain rule with:

$$\frac{\partial h_t}{\partial h_i} = \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-2}} ... \frac{\partial h_{i+1}}{\partial h_i} = \prod_{k=i}^{t-1} \frac{\partial h_{k+1}}{\partial h_k} \tag{2.38}$$

where, each term is the derivative of h_{k+1} with respect to h_k; which are given by

$$\frac{\partial h_{k+1}}{\partial h_k} = diag(f'(W_i x_i + W_c h_{i-1}))W_c \tag{2.39}$$

A vector is turned into a diagonal matrix by $diag$. Substituting the partial derivatives in 2.38 with the terms in 2.39 results in the back propagation gradient through k timesteps,

$$\frac{\partial h_k}{\partial h_1} = \prod_{i}^{k} diag(f'(W_i x_i + W_c h_{i-1}))W_c \tag{2.40}$$

The derivative $\frac{\partial h_k}{\partial h_1}$ expresses how the hidden state at time k changes when the hidden state is changed a small amount at time 1. The values of $f'(x)$ in 2.40 will always be less than 1 where, f' is the derivative of the activation function. This is because the activation function (tanh or sigmoid) maps all values into the range between -1 and 1. This means the derivative is bounded by 1. The derivative will reduces to 0 if the magnitude of the values of W_c are too small [Pascanu et al., 2013b]. When the gradient vanishes communication between hidden states get interrupted. Thus earlier hidden

states have no communication with later hidden states and failed to learn long term dependencies.

2.7.3 Long-Short-Term-Memory(LSTM)

The motivation behind developing LSTM was to remove the vanishing gradients issues that occur with RNN when processing long-term dependencies. The standard RNN consists of a chain of repeating modules of the neural network, where each module consists of a structure. For example, in Fig. 2.14 a single module is shown with a tanh layer. Such module structures are relatively simple. The LSTM has the same chain of repeating modules as an RNN except the LSTM module structure is relatively more complex. Each module consists of four layers rather a single layer as for an RNN module. Fig. 2.15 consists of a LSTM network architecture. The modules or memory blocks consist of an input gate, a forget gate, an output gate and the cell state. All these layers interact in a particular way. Information that will be added or removed to the cell state is controlled by three gates. An LSTM network computes a mapping from an input sequence $\vec{x} = (x_1, ..., x_t)$ to an output sequence $\vec{y} = (y_1, ..., y_t)$, where at first, values of input gate i_t and the candidate value of state of memory cell $\tilde{C}_t$ are calculated at time step t with,

$$i_t = \sigma(W_i\vec{x_t} + U_i\vec{h}_{t-1} + \vec{b_i}); \text{and} \tag{2.41}$$

$$\tilde{C}_t = \tanh(W_c\vec{x_t} + U_c\vec{h}_{t-1} + \vec{b_c}). \tag{2.42}$$

The input gate helps to store new information in the cell state. $\tilde{C}_t$ contains a vector of new candidate values created by each tanh layer to be added to the state. The forget gate value f_t is then calculated using,

$$f_t = \sigma(W_f\vec{x_t} + U_f\vec{h}_{t-1} + \vec{b_f}). \tag{2.43}$$

The forget gate helps to throw away information from the cell state and it also helps to reset the memory cells. The calculated values of i_t, $\tilde{C}_t$, f_t are then used to calculate the value of a new state of the memory cell C_t at time step t,

$$C_t = i_t \times \tilde{C}_t + f_t \times C_{t-1}. \tag{2.44}$$

The cell state works like a conveyor belt where information flows and it runs through the entire chain. Finally using the value of C_t, the output gate o_t value is calculated using,

$$\vec{o_t} = \sigma(W_o\vec{x_t} + U_o\vec{h}_{t-1} + V_oC_t + \vec{b_o}); \text{and} \tag{2.45}$$

$$\vec{h_t} = \vec{o_t} \times \tanh(C_t). \tag{2.46}$$

The output gate helps to compute the output using the cell state along with filtering the cell activations. In equations 2.41-2.46; where, W_i, W_c, W_f, W_o, U_i, U_c, U_f, U_o and V_o represent weight matrices. $\vec{b_i}$, $\vec{b_c}$, $\vec{b_f}$ and $\vec{b_o}$ represent bias vectors. σ is the logistic sigmoid function. Different combinations of these gates can be used to make it possible for the memory cells to deal with data with a longer horizon. The described LSTM structure does resolve the vanishing gradients issues and has been found to be suitable for modelling problems with long-term dependencies [Graves, 2012].

From (2.38) it can be seen that recursive derivative $\frac{\partial h_t}{\partial h_i}$ is the main reason to vanish the gradients. Long term dependencies could be learned if the derivative value does not go to 0. The main motivation of LSTM was to make this recursive derivative have a constant value so that it does not go to 0. Hochreiter & Schmidhuber [Hochreiter and Schmidhuber, 1997] in their LSTM formulation showed that the recursive gradient actually was equal to 1. To maintain this constant error flow, the gradient calculation was truncated. The full gradient of the recursive gradient is calculated below from the derivative $\frac{\partial C_t}{\partial C_{t-1}}$ and applying the multivariate chain rule.

$$\frac{\partial C_t}{\partial C_{t-1}} = \frac{\partial C_t}{\partial f_t}\frac{\partial f_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial i_t}\frac{\partial i_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial \widetilde{C_t}}\frac{\partial \widetilde{C_t}}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial C_{t-1}} + \frac{\partial C_t}{\partial C_{t-1}} \tag{2.47}$$

Where, C_t is a function of f_t ; f_t is the forget gate; i_t is the input gate; ∂C_t is the candidate cell state. These derivatives can be further expressed as:

$$\begin{aligned}\frac{\partial C_t}{\partial C_{t-1}} = C_{t-1}\sigma'(\cdot)W_f \times o_{t-1}tanh'(C_{t-1}) + \widetilde{C}_t\sigma'(\cdot)W_i \times o_{t-1}tanh'(C_{t-1}) \\ + i_t \tanh'(\cdot)W_C \times o_{t-1}tanh'(C_{t-1}) + f_t\end{aligned} \tag{2.48}$$

The difference between the recursive gradient of LSTM and unrolled RNN is that in unrolled RNN the terms $\frac{\partial h_t}{\partial h_{t-1}}$ will eventually take on a values that are always in the range 0 and 1 because of the activation function (tanh), and thus leads to the vanishing

gradient problem. Whereas, in LSTM the terms $\frac{\partial C_t}{\partial C_{t-1}}$ at any time step can consider values that are in the range 0 and 1. Moreover, the activations of the forget gate are greater than 0. Thus, for larger numbers of time steps there is a rarity that it will converge to 0. The values of f_t is set in such a way in order to bring the value of $\frac{\partial C_t}{\partial C_{t-1}}$ closer to 1. This helps to prevent the vanishing gradient problem during back propagation. LSTMs creates a connection between the forget gate activations and the gradients computation. This connection creates a path by which information flows through the forget gate and aids in modelling long term dependencies.

2.7.4 Bidirectional LSTM (BILSTM)

Bidirectional LSTM (BI-LSTM) works with a similar mechanism as bidirectional RNN except the data sequence is fed in both forward and backward directions using two separate hidden layers which are then connected to an output layer [Schuster and Paliwal, 1997]. The typical unrolled architecture of Bidirectional LSTM is shown in Fig. 2.16 which consists of a forward LSTM layer and a backward LSTM layer. The forward layer output sequence is calculated using inputs in a forward sequence from time $t-n$ to time $t-1$ while the backward layer output sequence is calculated using the reversed sequence from time $t-n$ to $t-1$. Outputs of both layers are calculated by using the standard LSTM equations (2.41) to (2.46).

An output vector $\vec{Y_n}$ is generated from forward and backward LSTM layers using the following equation 2.49:

$$\vec{Y_n} = \sigma(\vec{h}_n, \overleftarrow{h}_n). \tag{2.49}$$

where, n is the time step; σ is a function used to combine the outputs of the forward and backward LSTM layers. It can be a concatenating function, a summation function, an average function or a multiplication function.

2.7.5 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are feed-forward artificial neural networks. CNNs consist of alternating convolutional and subsampling layers, comparable with simple and complex cells in the human visual cortex [Yamins and DiCarlo, 2016, Mnih et al., 2015]. CNNs can be considered to mimic the human visual system and have

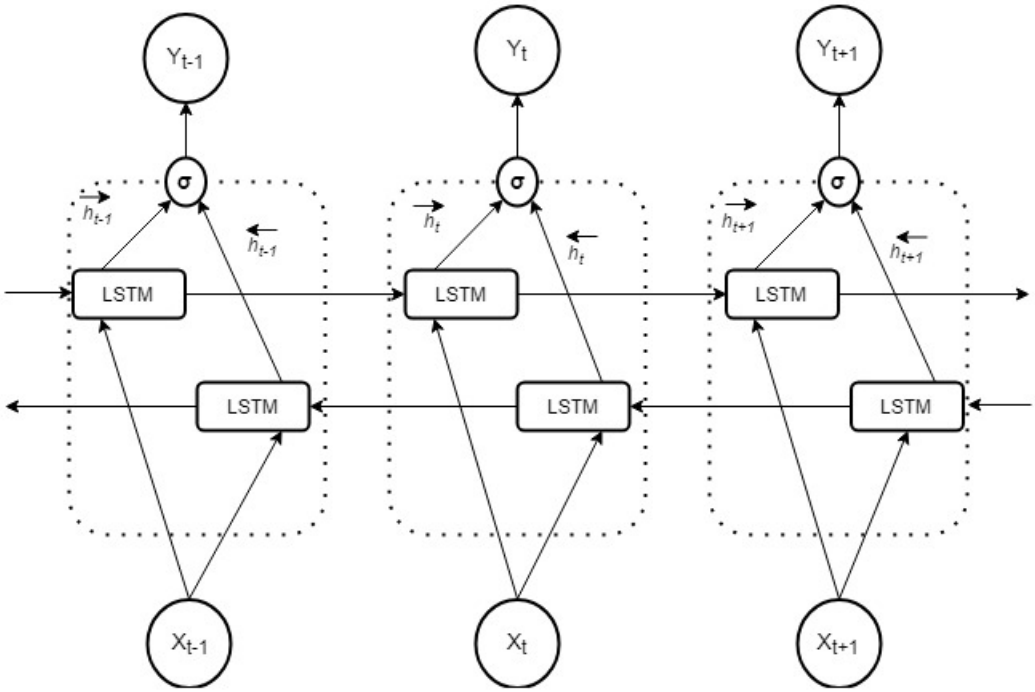


Figure 2.16: Unrolled architecture of Bidirectional LSTM where Y_n is the output, X_n in the input and σ is the combining function of two LSTM layer.

achieved subtle performance with recognizing patterns, structures and other functions such as tracking [Krizhevsky et al., 2012,LeCun et al., 2015]. CNN architecture consists of a set of kernels. To perform the convolution operation the kernels are moved across the input volume in a specified manner. The total number of free parameter is highly dependent on the kernel parameters.

A convolutional layer in the l^{th} layer can be computed as

$$X_k^l = f(\sum_c W_k^{(l),c} * X^{(l-1),c} + B_k^l) \tag{2.50}$$

where, k is the kernel number, c is the channel number of the input $X^{(l-1)}$, $W_k^{(l),c}$ is the k^{th} convolutional kernel corresponding to the c^{th} channel, and B_k^l is the learnable bias corresponding to the k^{th} kernel, $f()$ is the activation function and $*$ is the element wise multiplication.

A pooling layer is an important block in the CNN architecture. Which generally

placed between successive convolutional layers. This helps to reduce the parameters with the intuition that a rough location relative to other features is more important compared to the exact location. Max pooling and mean polling are the standard sub-sampling functions. Max pooling selects the maximum value from the kernels whereas mean polling takes the average of each kernel. Fully connected layers are then added to the structure for high-level reasoning. It helps to create a buffer between the learned features and the output by interpreting the features extracted from the convolutional layer. Back propagation is an approach that is used for training CNN.

In CNN an error term δ of the l^{th} layer can be defined as

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}).f'(z^{(l)}) \tag{2.51}$$

Where, $W^{(l)}$ is the parameter matrix of the lth layer, $z^{(l)}$ is the total weighted sum of inputs in layer l including the bias term can be written as $z^{(l)} = W^{(l-1)}a^{(l-1)} + b^{(l-1)}$, $a^{(l-1)}$ is the activation value in layer $l-1$ can be expressed as $a^{(l-1)} = f'(z^{(l-1)}$. The output layer (nl) is represented as $\delta^{(nl)} = \frac{\partial j}{\partial Z^{(nj)}}$ where j is the cost function.

The error term of the pooling layers are computed as

$$\delta_K^{(l)} = u((W_k^{(l)})^T \delta_K^{(l+1)}).f'(z_k^{(l)}) \tag{2.52}$$

where, k is the kernel number and $W_k^{(l)}$ is the kth kernel, $u()$ is the upsample operation which propagates the error through the pooling layer by calculating the error in regard of the incoming to the pooling layer.

The error term of the convolutional layers are propagated through as

$$\delta_K^{(l)} = conv(\delta_K^{(l+1)}, rot180(X_k^{(l+1)})).f'(z_k^{(l)}) \tag{2.53}$$

where, $conv()$ is the convolution operation and $rot180()$ performs 180 degrees rotation in order to achieve cross-correction of the convolution function.

CNNs were primarily developed for 2D signals. However 1D CNNs found useful for applications such as Electrocardiogram (ECG) classification [Kiranyaz et al., 2016], structural health monitoring [Abdeljaber et al., 2017] and motor-fault detection [Ince et al., 2016]. 1D CNNs have a very simple structure and can be trained with a limited amount of data compared to 2D CNNs. Moreover, 1D CNN effective enough to extract features from a fixed-length segment. CNN works the same way beyond its dimensions. It differs regarding the structure of input data and the way kernels or filters

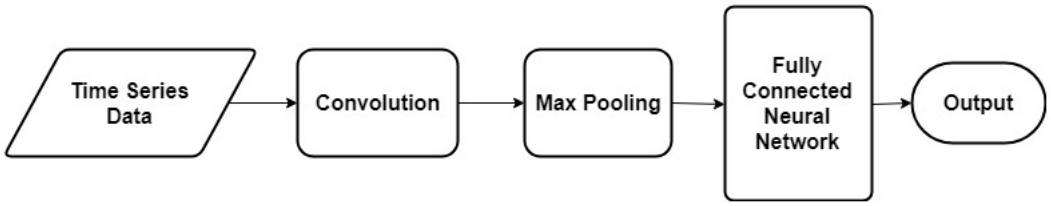


Figure 2.17: Generic architecture of the CNN

move across the data. 1D CNN also provides the luxury to use larger filter sizes and larger convolution windows. CNNs are a very good candidate in time series forecasting because of its filter feature extraction and composition ability. CNNs are also easier to train in comparison to RNNs because CNNs use convolution operation as opposed to recursion. The sliding window approach used with RNNs could also be used to train CNNs for time series forecasting.

2.8 Model Evaluation

Evaluation of the time series forecast model is required to evaluate the skill, capability, feasibility and estimation of a forecast model. The model evaluation procedure will help to inspect different aspects of a forecasting model such as different algorithm types, tuning parameters, and features [Raschka, 2015] [Chollet, 2018]. The design of an evaluation of a forecasting model depends on two tasks : model evaluation procedures and model evaluation metrics.

2.8.1 Model Evaluation Procedures

The model evaluation procedure is an integral part of the model development process and helps to explore how well a model will generalize to unseen or out-of-sample data. Data that has been used for training are not allowed to evaluate a forecast model. Using training data to evaluate model performance will result in an overoptimistic and overfitted outcome. In Machine learning and statistics, data are normally divided into two sets called training data and test data. The forecast model is fitted with training data and models make a prediction on test data. During this, there might be a scenario

where the model could be overfit or underfit the data. Such a scenario is unacceptable as this does not help models to generalize the predictions on unseen data. A model is said to overfit if the model is fit too closely to the training data. This is might happen when the model deals with a large number of variables and less number of samples or observations. More importantly, models learn the noise in the data rather learning the actual relationships between the features. Such a scenario will provide good training accuracy but will unable to generalize the prediction of unseen data. In underfitting, the model does not fit well with training data and thus unable to pick the available trends in the data. Underfitting, just right and Overfitting scenario are plotted in Fig. 2.18. Following this, a forecast model should avoid both scenarios. Different model evaluation procedure split the data into training and testing sets in ways to minimize over and underfitting. The design of an evaluation technique has an effect on model performance.

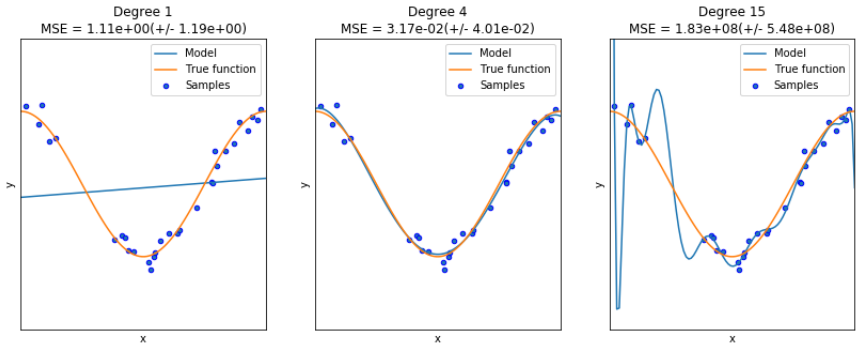


Figure 2.18: Example of underfit, just right and overfit scenario

2.8.1.1 Train and Test Split

Train test split model evaluation procedure divides the dataset into two separate set. The training set is used to train the model and the testing set is used to test the model. This model evaluation procedure allows to train and test the model with different data. It's also better in generalizing the unseen data. The downside of this model evaluation procedure is high variance.

2.8.1.2 Cross Validation

Cross-validation model creates a number of train-test splits and then it calculates the testing accuracy of each and finally averages them [Raschka, 2015] [Chollet, 2018] [Müller et al., 2016]. Fig. 2.19 shows a 5-fold cross-validation. Steps of cross-validation technique are given below:

- Divide the original data set into N equal parts.
- Take the first part as a testing set and the union of rest of the part as the training set.
- Calculate testing accuracy.
- Repeat the previous two steps for N times using different part as the training set.
- Calculate the average of all accuracy for final accuracy.

Cross-validation provides more out of sample accuracy because data is used more efficiently but it is N times slower than the train-test split. In general, the value of N is considered as 10. Cross-validation techniques can be improved by repeating the same process multiple times e.g. 1,000 iterations of a 10-fold cross validation. Another thing can be done to improve the performance is to hold out a set of data during the modelling process and then that hold outset is used as testing data, so this will provide a truly out of sample data and more a reliable performance.

2.8.2 Classification Evaluation Metrics

Classification metrics are used to evaluate classification problems of machine learning. To measure the performance of a prediction model with classification algorithms, researcher have used different evaluation metrics. Lee and mark on their work to evaluate the classification approach in investigating hypotensive event have used accuracy, AUC score, sensitivity and specificity as evaluation metrics [Lee and Mark, 2010a]. For, the same problem, Ghosh et al. [Ghosh et al., 2016] have used accuracy, sensitivity and specificity as evaluation metrics. These evaluation metrics have also been found to evaluate the performance of prediction model applicable for different others medical

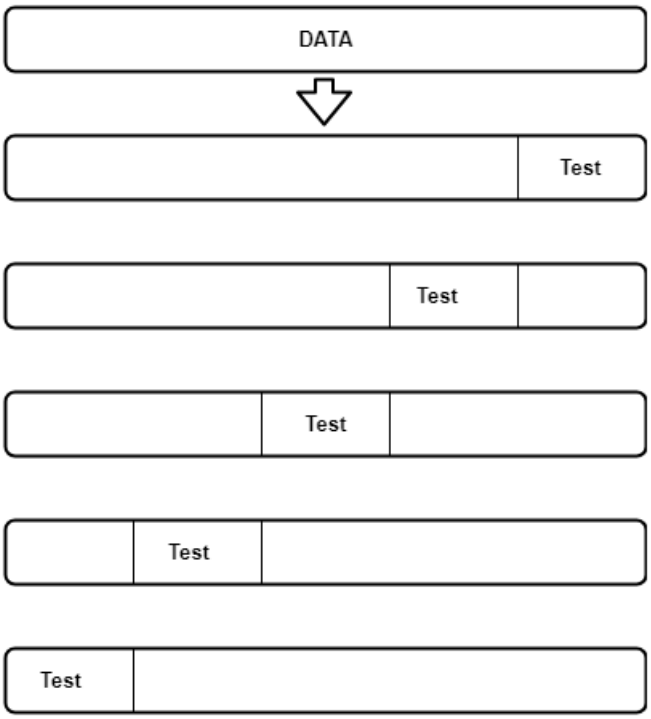


Figure 2.19: 5-fold cross-validation procedure

event prediction problem i.e. exacerbation events of pulmonary disease [Gather et al., 2002, van der Heijden et al., 2014], state estimation and clinical outcome [Marlin et al., 2012], dynamic mortality risk in the pediatric intensive care unit (PICU) [Aczon et al., 2017], predict mortality outcomes [Nguyen et al., 2017] [Zhu et al., 2018] [Nguyen et al., 2017] [Jo et al., 2017], diagnosis requirement in the pediatric intensive care unit (PICU) [Lipton et al., 2015b], predict heart failure [Isler, 2016] [Masetic and Subasi, 2016], predict cardiac arrest [Panahiazar et al., 2015, Kwon et al., 2018].

2.8.2.1 Confusion Matrix

A confusion matrix helps to describe classification model performance with details information. Information are presented in a table which presents details of correct and incorrect predictions [Müller et al., 2016]. The shape of confusion matrix is AxA, where A represents the classes. Table 2.13 displays a confusion matrix which consists of two

target values (positive and negative). Where:

- TP (True Positive) = Positive target is correctly predicted
- TN (True Negative) = Negative target is correctly predicted
- FP (False Positive) = Negative target is wrongly predicted
- FN (False Negative) = Positive target is wrongly predicted

Table 2.13: Confusion matrix details

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	TP	FP	Positive Predictive Value	TP/(TP+FP)
	Negative	FN	TN	Negative Predictive Value	TN/(FN+TN)
		Sensitivity	Specificity		
		TP/(TP+FN)	TN/(FP+TN)		

Table 2.14: Confusion matrix details with example

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	60	30	Positive Predictive Value	0.66
	Negative	40	70	Negative Predictive Value	0.63
		Sensitivity	Specificity		
		0.60	0.70		

- Classification Accuracy : One of the most common approaches to evaluating a forecast model is to calculate the percentage of correct predictions also known as accuracy. Classification accuracy consider number of correct predictions of all predictions. Such a metric is easiest to understand and calculate.

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.54}$$

- Positive Predictive Value(PPV) or Precision : the proportion of positive cases that were correctly predicted.

$$PPV = \frac{TP}{TP + FP} \tag{2.55}$$

- Negative Predictive Value(NPV): the proportion of negative cases that were correctly predicted.

$$NPV = \frac{TN}{FN + TN} \tag{2.56}$$

- Sensitivity or Recall : the proportion of actual positive cases which are correctly predicted.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.57}$$

- Specificity : the proportion of actual negative cases which are correctly predicted.

$$Specificity = \frac{TN}{TN + FP} \tag{2.58}$$

- False positive rate : 1 - specificity

Table 2.14 represents an example of a confusion matrix. Confusion matrix provides a complete picture of how the classifier is performing and aid in model selection.

2.8.2.2 ROC Curves and Area Under ROC Curve

ROC stands for Receiver Operating Characteristic and is considered as a visual tool for evaluating classification model. The ROC curves plot false positive rate on X-axis and sensitivity on Y-axis. A random model will provide a diagonal line on the ROC curve. The high the lines climb towards top-left means the model predicted the classification problem with high accuracy. An example of a receiver operating characteristic is plotted in Fig. 2.20.

Area Under ROC Curve (AUROC) is also used to evaluate the classification models. AUROC is also known as c-index [Cook, 2008]. According to c-index, AUROC can have ranged from .5 to 1. 1 is considered as perfect prediction ability whereas .5 indicates no predictive ability. The range between 0.700-0.800 is considered as reasonable and a range exceeding 0.800 is classified as good.

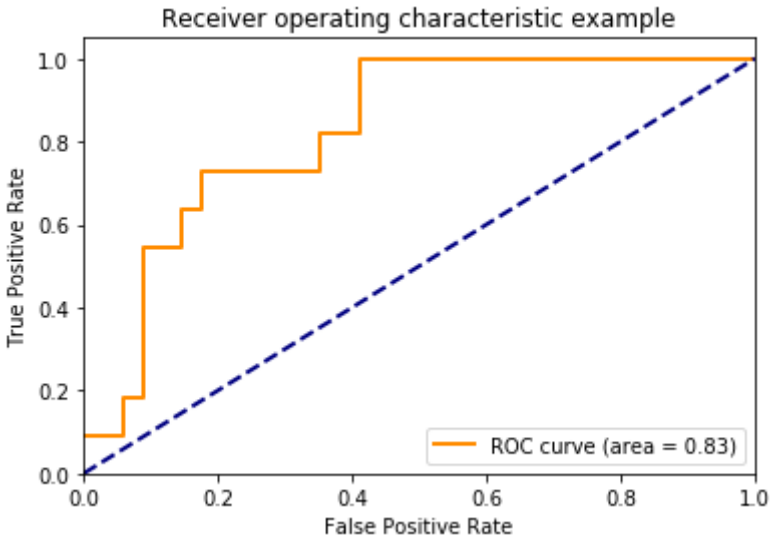


Figure 2.20: Example of receiver operating characteristic(ROC) with Area Under Curve (AUC) equal to 0.83

2.8.3 Regression Metrics

Regression metrics evaluate a regression model's capability to predict. Regression metrics do this by comparing the difference between forecasted and actual values. In general, the difference between the predicted and actual value is known as the residual. Residuals can be computed for every point in the target window. This plays a vital role in exploring the effectiveness of a forecast model. Small residuals suggest the model is producing a good forecast, while if the collection of residuals are large then it means the model is producing a poor forecast. However, using residuals is not meaningfull if there is a large number of data points at the target window. For such scenarios, statisticians have developed summary measurement methods that will consider a large number of residuals. Then all residuals are summarised into a single value which shows how a forecast model is performing. There are several evaluation metrics to measure regression forecast model performance. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are the most common metrics for this purpose. In specific to physiological time series forecasting, researchers have used RMSE and MAE as evaluation metrics. Lee and mark on their work to evaluate the results of 30 minutes continuous blood pressure with a neural network architecture have used MAE

as evaluation metrics [Lee and Mark, 2010a]. Billis and Bamidis have used RMSE and MAE as evaluation metrics to evaluate the forecast results of simulated blood pressure time series data [Billis and Bamidis, 2014]. In the most recent work by Li et al. to evaluate forecast results of blood pressure with RNNs, they have used RMSE and MAE [Li et al., 2017]. Sideris et al. [Sideris et al., 2016] and Su et al. [Su et al., 2018] used RMSE as an evaluation metric in forecasting physiological time series data.

2.8.3.1 Mean Absolute Error

The Mean Absolute Error (MAE) calculates the residual of every data point in the target window while considering the absolute value of each point. MAE then describes the magnitude of the residuals by taking the average of the all residuals. MAE can be expressed and calculated using :

$$\bar{e} = \frac{\sum_{i=1}^{m} |y_i - \widehat{y}_i|}{m} \tag{2.59}$$

where y_i are the actual values, $\widehat{y}_i$ are forecast values and m is the number of target output data.

The limitation of MAE is that it does not help identify whether the model has under-performed or over-performed as the absolute value of the residual is used. Large MAE means model predicts the outcome with more error whereas an MAE of 0 means that the model predicted the outcome perfectly.

2.8.3.2 Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) represents model accuracy as a percentage. MAPE can be expressed and calculated using :

$$\bar{e} = \frac{100\%}{m} \sum_{i=1}^{m} \left| \frac{y_i - \widehat{y}_i}{y_i} \right| \tag{2.60}$$

where y_i are the actual values, $\widehat{y}_i$ are forecast values and m is the number of target output data. The difference between the actual and forecast values are divided by the actual values. This is considered for every target points m and their absolute value is summed. The outcome is further divided by the number of target points m and finally multiplied by 100 to make it a percentage error.

2.8.3.3 Mean Squared Error

The Mean Squared Error (MSE) calculates the residual of every data points in the target window while considering the square difference of each point. MSE then describes the magnitude of the residuals by taking the average of the all residuals. MSE can be expressed using :

$$\bar{e} = \frac{\sum_{i=1}^{m} (y_i - \widehat{y}_i)^2}{m} \tag{2.61}$$

where y_i are the actual values, $\widehat{y}_i$ are forecast values and m is the number of target output data.

MSE is similar to MAE, except it squares the differences before summing them, instead of using the absolute value. However, MSE will provide larger values compared to MAE as the difference is squared. The model will be penalized more by MSE than MAE as the error grows quadratically in MSE compared to proportionally in MAE.

2.8.3.4 Root Mean Squared Error

Root Mean Squared Error (RMSE) is the square root of the MSE. RMSE can be expressed using :

$$\bar{e} = \sqrt{\frac{\sum_{i=1}^{m} (y_i - \widehat{y}_i)^2}{m}} \tag{2.62}$$

where y_i are the actual values, $\widehat{y}_i$ are forecast values and m is the number of target output data.

The RMSE produces a comparatively high weight to large errors by taking the square root of the average squared errors. The RMSE can be more effective for many applications where the large error is particularly undesirable. The advantage of RMSE over MAE is that RMSE does not consider the absolute value.

All of the above regression metrics calculate the performance of a model from the residuals found from the predicted and actual values. The magnitude of the metrics decides the model performance. Small values indicate good forecast and large values suggest poor forecast.

2.9 Summary and Concluding Remarks

This chapter described the concepts that are related to the research problems in this thesis and their solutions. Types of time series forecasting and strategies required are described. Supervised machine learning and its algorithms have also been introduced. Data processing and transformation techniques for supervised machine learning are mentioned. Details of evaluation procedures and metrics of machine learning models have also been described.

Chapter 3

Literature Review

This chapter aims to provide a concise overview of the research on time series analysis of physiological data in solving various problems in the medical domain. Prediction of physiological time series data (time series forecasting and classification) using machine learning techniques could be used to help intervene in dangerous clinical events. The review identifies some of the applications and limitations of classification based approaches applied to physiological time series for event prediction. The study then describes how regression based approaches can overcome the problem and also covers the shortcomings of regression based approaches involved in forecasting physiological time series data. The review also includes how current literature have so far been unable to consider different aspects of machine learning algorithms and techniques in forecasting physiological time series for medical applications. The review details which regression based approaches have found to be useful for various medical applications. In particular, the role and limitations of analysing physiological time series data in exploring hypotensive and bradycardia events are also identified.

3.1 Prediction of Physiological Time Series Data

Physiological signals have been extensively used in the medical sector for predicting different events. Prediction includes both time series classification and time series forecasting.

3.2 Time Series Classification of Physiological Time Series Data

Time series classification of physiological signals have been used for the prediction of a number of different types of clinical events. This has included prediction of dialysis in critically ill patients', mortality risk predictions in paediatric critical care, hypotensive episode predictor for intensive care and prediction of morbidity of tuberculosis [Ongenae et al., 2013, Eswaran et al., 2010, Aczon et al., 2017, Lee and Mark, 2010a]. Time series Heart Rate (HR) and systolic, diastolic and Mean Arterial Blood Pressure (MAP) data, pulse pressure, and relative cardiac output were extracted from Multiparameter Intelligent Monitoring in Intensive Care(MIMIC) II database by Lee and Mark. These have been used for the prediction of hypotensive episodes in ICU using a 3-layer artificial Neural Network (NN) [Lee and Mark, 2010a]. They considered time series classification to correctly classify hypotensive events versus non-hypotensive events. Lehmani et al. used the vital-sign time series of HR and MAP from MIMIC II database along with the simplified acute physiology score for classifying the patients' with sepsis [Li-wei et al., 2014]. Ghosh et al. performed time series classification to predict acute hypotension risk at ICU. They extracted symbolic sequences of blood pressure (BP) time-series data from MIMIC II database. They used a symbolic aggregate approximation method to convert time series data to symbolic sequence [Ghosh et al., 2016]. Ghassemi et al. have used vital sign data, along with clinical notes [Ghassemi et al., 2015]. Cerebrovascular pressure reactivity and mortality prediction were performed. They considered multivariate time series modelling with Gaussian processes [Ghassemi et al., 2015]. Time series classification was performed to predict two class of clinical outcomes (in-hospital mortality and 1-year post-discharge mortality). Time series of physiological data from the MIMIC II clinical dataset was used by Xue et al. to predict 30 days ICU readmissions [Xue et al., 2018]. Wu et al. considered multidimensional physiological time series data from the MIMIC II. They predicted vasopressor administration and weaning in the intensive care unit [Wu et al., 2017]. Physiological time series data and electronic medical records were used to predict sepsis in the ICU. Sepsis is considered to be one of the leading reasons for mortality, morbidity and cost overruns in critically ill patients' [Shashikumar et al., 2017]. Ghosh et al.

used sequential patterns extracted from MAP, HR, and Respiratory Rate (RR) signals for septic shock patients' from the MIMIC II database [Ghosh et al., 2017]. Forkan et al. used patterns of multiple vital signs (e.g. HR, BP) from a large number of similar patients'. They built a prognostic model that can accurately identify dangerous clinical events of a home-monitoring patient in advance using knowledge learned from the patterns [Forkan et al., 2017]. Johnson et al. extracted vital sign measurements (heart rate, blood pressure, respiratory rate, oxygen saturation) for mortality prediction of ICU from MIMIC III [Johnson et al., 2017]. Kwon et al. used systolic BP, HR, RR, and body temperature to predict cardiac arrest [Kwon et al., 2018].

3.2.1 Limitations and Challenges

Physiological times series prediction in medical prediction problems has been mostly conducted by time series classification. However, time series classification is typically limited to short-term and single-step prediction. In time series classification approach, prediction model only predicts the next value. Predicting multiple values in future or long term forecasting is an integral part of clinical decision making, as has been shown in glucose monitoring [Mclachlan et al., 2007] and Electroencephalogram (EEG) monitoring in ICU [Vespa et al., 1999]. Long term forecasting is useful where the field of application requires long-term duration forecasting. This class of prediction is very challenging, and there is a lack of studies available consisting of machine learning algorithms and methodologies [Weigend, 2018, Masum et al., 2018, Masum et al., 2017]. Moreover, most of the current time series classification approaches directly map input physiological signals to output values. Thus, they are unable to model underlying temporal dependencies in physiological data dynamics. From the above-described literature, it is quite clear that physiological time series data plays a vital role in medical prediction problems. The MIMIC database seems to be one of the best resources of these physiological time series data as researchers have frequently used this source.

Table 3.1: Summary of classification approaches in analysis of physiological time series data.

Approach	Time Series Classification
Application	**Dialysis in critically ill patients'** [Ongenae et al., 2013]. **Mortality risk predictions in paediatric critical care** [Aczon et al., 2017, Johnson et al., 2017]. **Hypotensive episode predictor for intensive care** [Lee and Mark, 2010a, Ghosh et al., 2016]. **Prediction of morbidity of tuberculosis** [Eswaran et al., 2010]. **Sepsis** [Shashikumar et al., 2017, Ghosh et al., 2017, Wu et al., 2017]. **Cerebrovascular pressure reactivity and mortality prediction** [Ghassemi et al., 2015]. **ICU readmissions** [Xue et al., 2018]. **Cardiac arrest** [Kwon et al., 2018].
Data	**Heart Rate** [Forkan et al., 2017, Kwon et al., 2018, Johnson et al., 2017] [Lee and Mark, 2010a, Li-wei et al., 2014, Xue et al., 2018, Ghosh et al., 2017]. **Blood Pressure** [Kwon et al., 2018, Johnson et al., 2017, Ghosh et al., 2016] [Forkan et al., 2017, Lee and Mark, 2010a, Li-wei et al., 2014]. **Pulse pressure** [Lee and Mark, 2010a, Xue et al., 2018]. **Clinical notes** [Ghassemi et al., 2015, Wu et al., 2017]. **Respiratory rate** [Kwon et al., 2018, Johnson et al., 2017] [Lee and Mark, 2010a, Xue et al., 2018, Ghosh et al., 2017, Ghosh et al., 2017]. **Oxygen saturation** [Johnson et al., 2017] [Lee and Mark, 2010a, Xue et al., 2018]
Machine Learning Algorithms	**Multi-layer perceptrons** [Narin et al., 2014, Elfadil and Ibrahim, 2011] [Jovic and Bogunovic, 2011, Isler, 2016]. **KNNs** [Masetic and Subasi, 2016, Mani et al., 2014, Isler, 2016, Narin et al., 2014]. **Random forests** [Masetic and Subasi, 2016, Austin et al., 2013] [Guidi et al., 2015, Guidi et al., 2014, Zolfaghar et al., 2013] [Vedomske et al., 2013, Panahiazar et al., 2015, Kwon et al., 2018]. **Logistic regression** [Wu et al., 2010, Zolfaghar et al., 2013] [Zhai et al., 2014, Churpek et al., 2016, Panahiazar et al., 2015, Kwon et al., 2018]. **Decision trees** [Son et al., 2012, Pocock et al., 2005]. **Neural network** [Lee and Mark, 2010a, Aczon et al., 2017, Lipton et al., 2015b]. **Feedforward networks** [Lasko et al., 2013, Che et al., 2015].

Table 3.2: Strengths and weaknesses of classification approaches in analysis of physiological time series data.

Approach	Time Series Classification
Strengths	**Effective in classifying binary class problems** [Lee and Mark, 2010a] [Lee and Mark, 2010a, Ghosh et al., 2016, Ghassemi et al., 2015] [Lasko et al., 2013, Lipton et al., 2015b]. **Helps healthcare professionals in decision making** [Li-wei et al., 2014] [Lee and Mark, 2010a, Ghosh et al., 2016, Ghassemi et al., 2015] [Wu et al., 2017, Lasko et al., 2013, Lipton et al., 2015b] [Johnson et al., 2017, Kwon et al., 2018]. **Helps healthcare policy makers** [Xue et al., 2018] [Shashikumar et al., 2017, Zolfaghar et al., 2013, Vedomske et al., 2013].
Weaknesses	**Predicts only singe step or the next value** [Li-wei et al., 2014] [Ghosh et al., 2016, Ghassemi et al., 2015, Johnson et al., 2017] [Kwon et al., 2018, Lipton et al., 2015b, Carlin et al., 2018]. **Unable to provide insight into clinical events** [Ghosh et al., 2016] [Forkan et al., 2017, Kwon et al., 2018, Guidi et al., 2015] [Panahiazar et al., 2015, van der Heijden et al., 2014]. **Some cases avoid underlying temporal dependencies in time series** [Weigend, 2018, Masum et al., 2018]. **Hyperparameter tuning was not considered in many cases** [Lee and Mark, 2010a, Carlin et al., 2018, Jo et al., 2017, Nguyen et al., 2017] [Zhu et al., 2018, Lipton et al., 2015b, Kwon et al., 2018].

3.3 Time Series Forecasting of Physiological Time Series Data

Continuous monitoring is often a crucial part of clinical decision making, as has been shown in glucose monitoring [Mclachlan et al., 2007] and EEG monitoring in the ICU [Vespa et al., 1999]. However, there is a minimal number of work forecast continuous values of physiological data.

Lee and mark forecast 30 minutes of continuous blood pressure [Lee and Mark, 2010a]. They used an artificial neural network architecture and achieved a best mean

absolute error of 9.67 percent [Lee and Mark, 2010a]. Lee and mark extracted multi-dimensional hemodynamic data as features from the MIMIC 2 database. These data were analysed to predict the hypotensive events 1 or 2 hours in advance. They showed that multiple compilations perform better than a single compilation. They also showed that an increase in gap size reduces forecast performance, and window size has little impact on forecast performance. Moreover, they found, there are cases where regression outcome considerably differs from the actual result. However, they were unable to draw any conclusions as to how to utilise the forecast outcome in event prediction. Hyperparameter tuning of the architecture was missing. The work also did not consider the comparison of forecast strategies and data approach on their structure.

Billis and Bamidis forecast artificial blood pressure time series data using ARIMA, SVM, neural networks and a gaussian process based forecasting algorithm [Billis and Bamidis, 2014]. There was a minimal number of data considered for training and testing purpose. Only ten test scenarios were considered. Hyperparameter tuning was also not considered in their work.

Li et al. predicted blood pressure using RNNs with a contextual layer [Li et al., 2017]. They considered sequential data of variable length and features extracted from contextual data to predict numerical values of BP. They showed that adding a contextual layer slightly improves the model accuracy. The best result gave MAE and RMSE vlaues of 3.2291 and 4.2681 respectively. An LSTM along with contextual layer were used. However, it is interesting to note that without the contextual layer, MAE and RMSE values of 3.2639 and 4.3168 were achieved. Moreover, the authors found that the LSTM algorithm had an advantage over RNN, MLP, SVM, GBRT and factorization machine. Similar to previous work, their work also did not show how the forecasted outcome can be utilised in medical applications. Moreover, their work does not use hyperparameter tuning of the chosen models. Comparison of forecast strategies also was not considered.

Sideris et al. [Sideris et al., 2016] predicted continuous ABP using LSTM models. They extracted ABP data from the MIMIC database and used 42 patients' data to validate their model. The LSTM model achieves an average, maximum and minimum RMSE values of 6.042, 22.850 and 2.448 respectively in predicting continuous ABP. Similar to previous work, hyperparameter tuning and comparison of forecast strategies

were not performed.

In more recent work by Su et al. [Su et al., 2018], predicted Systolic BP (SBP) and Diastolic BP (DBP) sequence using a multi-layer BI-LSTM network. They used static and a multi-day continuous BP dataset extracted from 84 and 12 healthy people. Sampling frequency for the data was 1000 Hz. The proposed model achieved an RMSE values of 3.90 and 2.66 for systolic and diastolic blood pressure. They showed that the proposed model outperformed traditional regression models, i.e. support vector regression, decision tree, bayesian linear regression and kalman filtering. Moreover, it appeared to show that a prediction model that takes into account temporal dependencies in BP dynamics improves prediction accuracy significantly.

3.3.1 Limitations and Challenges

Accurate forecasting of physiological time series data could be meaningful to specific medical applications. Existing works predict the continuous value of physiological time series data without considering different forecast strategies. In contrast to this, Ben Taieb et al. considered some non-medical data [Taieb et al., 2012]. They described different forecast strategies and showed that forecast strategies play a vital role in long-term forecasting scenario [Taieb et al., 2012]. Another consideration can be the form of the data. Aboagye-Sarfo et al. and Preez et al. worked with hospital emergency data and international tourism data respectively. Their work showed the importance of different data approaches in forecasting by comparing univariate and multivariate data [Du Preez and Witt, 2003] [Aboagye-Sarfo et al., 2015]. Existing literature that forecasts physiological time series data have not performed comparisons of these two data approaches. It has also been found that a lack of collaborations between these different fields is creating a barrier to further developments. Investigating the forecast strategies and different data approaches in predicting physiological time series data would explore the efficiency of forecast strategies and data approach.

Table 3.3: Summary of regression approaches in analysis of physiological time series data.

APPROACH	Time Series Forecasting
Application	**Predict events** [Lee and Mark, 2010a, Ghosh et al., 2016]. **Forecast artificial blood pressure** [Billis and Bamidis, 2014]. **Predict the numerical values or sequence of BP** [Li et al., 2017, Sideris et al., 2016, Su et al., 2018].
Data	**Heart Rate** is used to forecast BP [Lee and Mark, 2010a, Su et al., 2018, Li et al., 2017]. **Blood Pressure** is used to forecast BP [Li et al., 2017, Sideris et al., 2016, Su et al., 2018] [Lee and Mark, 2010a, Ghosh et al., 2016, Billis and Bamidis, 2014]. **Pulse pressure** is used to forecast BP [Lee and Mark, 2010a]. **Clinical notes** is used to forecast BP [Lee and Mark, 2010a]. **Respiratory rate** is used to forecast BP [Lee and Mark, 2010a]. **Oxygen saturation** is used to forecast BP [Lee and Mark, 2010a]. **Contextual information** is used to forecast BP [Li et al., 2017].
Machine Learning Algorithms	**RNN** [Lee and Mark, 2010a, Li et al., 2017]. **ARIMA** [Billis and Bamidis, 2014]. **NN** [Billis and Bamidis, 2014]. **ANN** [Lee and Mark, 2010a]. **LSTM RNN** [Li et al., 2017, Sideris et al., 2016]. **BILSTM RNN** [Su et al., 2018].
Event Forecast	**Applied to predict hypotensive event** [Lee and Mark, 2010a].

3.4 Machine Learning in Physiological Time Series Classification

Researchers have used a wide variety of machine learning techniques for physiological time series classification. These include multi-layer perceptrons [Narin et al., 2014, Elfadil and Ibrahim, 2011, Jovic and Bogunovic, 2011, Isler, 2016]; KNNs [Isler, 2016, Narin et al., 2014, Masetic and Subasi, 2016]; random forests [Masetic and Subasi, 2016, Austin et al., 2013]; logistic regression [Wu et al., 2010] and decision trees [Son et al., 2012] to predict heart failure. Decision trees [Pocock et al., 2005] and random forests [Guidi et al., 2015, Guidi et al., 2014] have been used for the prediction of destabilizations. Re-hospitalisations have been predicted using logistic regression [Zolfaghar et al., 2013]; and random forests [Zolfaghar et al., 2013, Vedomske et al., 2013]. Panahi-

Table 3.4: Strengths and weaknesses of regression approaches in analysis of physiological time series data.

Approach	Time Series Forecasting
Strengths	**Multi-step prediction or long term forecasting** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018]. **Contribution to the decision-making process** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018]. **Contribution to the predictive health care** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018].
Weaknesses	**Lack of investigation and comparison of machine learning algorithms** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018]. **Comparison of forecast strategies and data approaches were not considered** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018]. **Cross-validation scenario was not considered in model evaluation** [Billis and Bamidis, 2014, Su et al., 2018] [Li et al., 2017, Sideris et al., 2016]. **Hyperparameter tuning of the architecture was missing** [Lee and Mark, 2010a, Billis and Bamidis, 2014] [Li et al., 2017, Sideris et al., 2016, Su et al., 2018]. **The forecast outcome was not utilised in medical applications** [Billis and Bamidis, 2014, Sideris et al., 2016] [Li et al., 2017, Su et al., 2018].

azar et al. and Kwon et al. have used logistic regression and random forests to predict cardiac arrest [Panahiazar et al., 2015, Kwon et al., 2018]. KNNs, along with other techniques, have been used to develop non-invasive predictive models for late-onset neonatal sepsis [Mani et al., 2014]. Off-the-shelf medical data and Electronic Health Records (EHRs) were used by Mani et al. [Mani et al., 2014]. Zhai et al. have used logistic regression to predict the need for paediatric ICU transfer within the first 24 hours of admission [Zhai et al., 2014]. They have analysed different temporal measurements of EHRs of 7,298 patients'. A ten-fold cross-validation technique was used to evaluate

the model [Zhai et al., 2014]. Churpek et al. used logistic regression models to predict vital signs, laboratory values, and demographic variables [Churpek et al., 2016]. These were then used to predict ICU transfer, cardiac arrest and death [Churpek et al., 2016]. Ten-fold cross-validation was used to evaluate the model. Dynamic bayesian networks have been used to find the conditional dependence structure of physiological variables and predict exacerbation events of pulmonary disease [Gather et al., 2002, van der Heijden et al., 2014]. The study used a confusion matrix and AUC-ROC to evaluate the performance model. Switching vector autoregressive technique was applied to several physiological variables by Li-wei et al. for predicting state estimation and clinical outcome [Li-wei et al., 2014]. This study used a train test split and cross-validation as the model evaluation procedure. Accuracy, AUC-ROC and recall were used as model evaluation metrics. Aczon et al. used a recurrent neural network for predicting dynamic mortality risk in the Paediatric Intensive Care Unit (PICU). They found that a recurrent neural network shows better performance compared to clinically used scores and static machine learning algorithms [Aczon et al., 2017]. Vital signs were used along with other clinical information in the study. The study used a train and test data split technique to evaluate the model along with ROC-AUC as the model evaluation metrics. Neural networks have been applied to different types of medical data to analyse different medical problems for more than 20 years [Lipton et al., 2015b]. Feedforward networks with the sliding window technique have been applied to medical time series to classify cases of gout, leukemia [Lasko et al., 2013] and critical illness [Che et al., 2015].

3.4.1 Limitations and Challenges

The above mentioned machine learning techniques were mainly used for time series classification in the medical domain. It has been shown that machine learning algorithms can be applied to time series prediction of physiological time series. Moreover, all these techniques are not specifically designed to deal with temporal data or long-term dependencies. Machine learning-based sequential techniques such as Gaussian processes, Hidden Markov Models (HMMs), Conditional Random Fields (CRFs) are better suited for temporal data [Hill et al., 1996, Baccouche et al., 2011, Lafferty et al., 2001]. But they are unable to handle long-term dependencies. Long Short-Term Memory (LSTM)

are a class of RNNs. They are specifically designed for sequential data and have been found to be very effective in time series prediction [Hochreiter and Schmidhuber, 1997]. LSTM and BI-LSTM RNN have also been applied to medical time series data. Lipton et al. used LSTM networks to assign diagnostic information learnt using multivariate time series of clinical measurements [Lipton et al., 2015b]. Nguyen et al. used both LSTM and BI-LSTM RNN models to predict mortality outcomes of patients' in ICUs by modelling physiological time-series data [Nguyen et al., 2017]. Zhu et al. considered supervised Bidirectional LSTM Networks to predict ICU mortality [Zhu et al., 2018]. RNNs and LSTM have been used by [Carlin et al., 2018] and [Nguyen et al., 2017] to process the time series variables in ICU. LSTM have also been used to classify the diagnosis required for patients' using EHRs in the Pediatric Intensive Care Unit (PICU) [Lipton et al., 2015b]. BI-LSTM are used to predict mortality outcomes in ICU [Nguyen et al., 2017] [Jo et al., 2017]. LSTM have used outlier's detection in time series 23 abnormal and normal behaviours in ECG time series [Chauhan and Vig, 2015]. LSTM have been used by Lipton et al. for recognising patterns from EHRs dataset. The dataset consisted of time series data of different clinical measurements such as body temperature, heart rate, diastolic and systolic blood pressure. The study showed that LSTM is effective enough to deal with long-range dependencies and non-linear characteristics of the data [Lipton et al., 2015b]. However, these RNNs (LSTM and BI-LSTM) were used for medical applications where only the next value is predicted, thus limited to short-term prediction. Predicting multiple values for the future for medical applications could open up medical applications even further.

3.5 Machine Learning in Physiological Time Series Forecasting

A set of artificial time-series data for forecasting of pathological signs was investigated by Billis Bamidis [Billis and Bamidis, 2014]. They have used time-series forecasting techniques such as exponential smoothing, Box-Jenkins seasonal Auto-Regressive Integrated Moving Average (ARIMA) models and neural networks. An artificial neural network architecture was used by Lee and mark to forecast 30 minutes of continuous blood pressure [Lee and Mark, 2010a]. LSTM RNNs, along with a contextual layer,

was used by Li et al. to forecast blood pressure. They compared LSTM RNNs with RNN, MLP, Support vector machine, Gradient boosted regression tree, Factorization machine and found that LSTM RNNs performs better [Li et al., 2017]. Sideris et al. [Sideris et al., 2016] used LSTM algorithms to forecast continuous ABP. Su et al. used a multi-layer BI-LSTM network in forecasting Systolic BP (SBP) and Diastolic BP (DBP) sequence [Su et al., 2018]. They showed that BI-LSTM networks outperformed traditional regression models. Comparisons included support vector regression, decision tree, bayesian linear regression and kalman filters.

3.5.1 Limitations and Challenges

There is a minimal number of works that consider machine learning algorithms in forecasting physiological data. The few works that have considered machine learning algorithms in forecasting physiological data have had many limitations. Hyperparameter tuning was not considered in [Billis and Bamidis, 2014] [Lee and Mark, 2010a] [Li et al., 2017] [Sideris et al., 2016] [Su et al., 2018]. However it is believed that hyperparameter tuning is essential when a forecast model is considered for any real-life application.

Combination of machine learning algorithms have been applied to a wide range of different applications. In particular, Kim et al. [Kim and Kim, 2019] used a combination of CNN and BI-LSTM algorithm to forecast stock prices. They showed that the feature fusion capability of a CNN-BILSTM based approach outperforms CNN or LSTM individually. Huang et al. combined CNN and LSTM for forecasting particulate matter in smart cities [Huang and Kuo, 2018]. They compared CNN-LSTM algorithm with traditional algorithms like support vector machine, decision tree, MLP, CNN, and LSTM. They found that the CNN-LSTM based system performed the best. A hybrid neural network model consisting of CNN and LSTM was used by Kim et al. for power demand forecasting [Kim et al., 2019]. Their hybrid model performed better than ARIMA and LSTM algorithm individually. These aforementioned of CNNs and LSTMs or BILSTMs combinations have achieved some good results. However, the combination of machine learning algorithms in forecasting physiological time series data is yet to be explored.

3.6 Physiological Time Series Data in Predicting Critical Events

This research consider hypotensive and bradycardia events as critical events.

3.6.1 Hypotensive Events

A Hypotensive Episode (HE) is a life-threatening event. It can be recognised from hemodynamic monitoring often performed in the Intensive Care Units (ICU)s. If a patients' Mean Arterial Blood Pressure (MAP) drops below 60 mmHg for at least 90% of the 30 minutes, then that event is known as an HE [Irwin and Rippe, 2008]. The MAP is a measure of blood pressure which is calculated from a combination Systolic Pressure (SP) and Diastolic Pressures (DP) [Mann et al., 2014]

$$MAP = [2(DP) + SP]/3 \tag{3.1}$$

An HE can cause decreased tissue perfusion, cellular dysfunction and organ damage [Ghosh et al., 2016]. These risks begin to develop shortly after HE. Complications increase following the HE duration [Walsh et al., 2013]. Thus, healthcare professionals should be vigilant in detecting and treating HEs in advance. Predicting that an HE is imminent could allow healthcare professionals to prevent the clinical impact of HE. However, this is a big challenge for the reactive patient care paradigm which is currently in mainstream use. Existing early warning monitoring systems are not able to consider for this scenario. Current early warning systems consist of predefined clinical rules which are applied to vital sign data. However, the system raises the alarm reactively and in many cases, generates a significant number of false alarms [Pinsky, 2007, McGlynn et al., 2003]. Moreover, current systems do not take advantage of sequential information. Existing systems map the input physiological signals to output values without explicitly modelling underlying temporal dependencies. On the other hand, in many cases, it is almost impossible for a human clinician to analyse the continuous and quantitative complex medical data that is present in physiological time series.

Physiological time-series, specifically the patients' blood pressure time series feature, is used to predict the hypotensive event. In 2001, Bassale examined ABP wave-

forms; it was shown that there is a significant change in the ABP waveform shape 1 to 5 minutes before a hypotensive event [Bassale, 2001]. Bassale used the autocorrelation function of the blood pressure time series data. Crespo et al. also examined the patients' blood pressure time series. The variance of the ABP signal and the variance of the ABP wave's slope were used to predict hypotensive events [Crespo et al., 2002]. Moreover, Crespo et al. used the same size of the prediction window of 1 to 5 minutes as used by Bassale. In 2009 PhysioNet and Computers in Cardiology introduced an open challenge to encourage participants to develop a forecast model that predicts a hypotensive event [Moody and Lehman, 2009]. Researchers applied different techniques to analyse and predict hypotensive behaviours. These included neural networks, support vector machines, numerous statistical methods and hilbert-transform based techniques [Chen et al., 2009, Rocha et al., 2011, Langley et al., 2009, Ghaffari et al., 2010]. Limitations of such techniques are that the analysis has been conducted on a static or absolute measure of the blood pressure. This avoided the temporal dependency of the measured data. However, medical time series data has been found useful in recognising patterns [Lipton et al., 2015b]; discovering the pathological and physiological state of the patients' [Li-wei et al., 2014]; exploring physiological deterioration [Almeida and Nabney, 2016]; and predicting mortality outcomes [Nguyen et al., 2017, Zhu et al., 2018]. Another limitation of the forecast models mentioned above is that cross-dataset validation was not considered. Moreover, MIMIC II matched subset data have only several thousand physiological data records. However, machine learning techniques might require much more training data for effective forecasting. The more recently published MIMIC III waveform database matched subset is four times larger than the MIMIC II matched subset. This could be even more useful for machine learning techniques. Ghosh et al. [Ghosh et al., 2016] and Lee and Mark [Lee and Mark, 2010a] have extracted hemodynamic patterns from the MIMIC II matched subset. They have used blood pressure time series data to predict hypotensive events. Ghosh et al. with sequential contrast patterns, demonstrated the highest accuracy of 83.54% for single-mode dataset along with 60 min observation window and 60 min gap window [Ghosh et al., 2016]. The sensitivity and specificity were found to be 100% and 68.29% respectively. It was stated that the balanced nature of the single-mode datasets is responsible for a lower percentage of specificity. In another study, Lee

and Mark [Lee and Mark, 2010b] with neural network also demonstrated the highest accuracies of 76% for single-mode datasets extracted from MIMIC-II with 60 min observation window and 60 min gap window. The sensitivity and specificity were found to be 75% and 76% respectively. Ghosh et al. and Lee and Mark on their multimode datasets task included predictions of future episodes even if the patient is experiencing an ongoing episode. However, such information could be non-essential to healthcare professionals on the one hand. Furthermore, experiment with this type of information may not necessarily emulate the achievement of an applied system.

3.6.2 Bradycardia Events

Bradycardia is a clinical event caused due to slow heart rate. According to the national institutes of health, if the heart rate of an adult goes below 60 bits per minutes, then its called a bradycardia [Kusumoto et al., 2018]. Bradycardia could lead to heart failure, sudden cardiac arrest, stroke and mortality [Yong et al., 2015] [Cheung et al., 2015] [Kusumoto et al., 2018].

3.6.3 Limitations and challenges

Almost all these aforementioned works use classification algorithms to predict hypotensive events. Often, a simple binary classification is performed to predict a hypotensive event in advance. Limitations of classification in predicting a hypotensive event is that it does not help to explore or reflect the overall scenario of the hypotensive event. It just helps to identify whether there will be a hypotensive event or not in advance for a given time series. On its own, this could be useful to a healthcare professional. However, hypotensive events have a large range of blood pressures (between 0 to 59) in its definition. Thus, anything that is below 60 in a classification model would be predicted as a hypotensive event. According to this definition, 0 and 59 would both be considered as hypotensive events. However, it would obviously be different to healthcare professionals in terms of actions and treatments. In contrast to this forecasting of hypotensive events with regression algorithms could help to explore the differing nature of events along with the help of continuous MAP values. This could allow a healthcare professional to have access to additional information. Furthermore, it could

further help in the decision making process. Moreover, existing works have difficulties in dealing with temporal dependencies in physiological time series.

Some works consider a minimal number of samples for training and testing. The ratio of classes in all the datasets were unbalanced. All the works extract data following the MAP range on HE definition. MAP range varies from 0 to 59 following HE definition. However, as the dataset was not shared, it is quite hard to understand what ranges they have actually considered for classes. It is also quite hard to compare the results with the existing work because of a standard dataset. PhysioNet and Computers in Cardiology challenge in 2009 provides a dataset for this problem. However, the dataset has limited data to work on when considering machine learning. Moreover, datasets shared for the HE problem by Ghosh et al. [Ghosh et al., 2016] was found to be depreciative.

A few studies concentrated on risk factors in developing bradycardia and management of bradycardia [Yorozu et al., 2007] [Pollard, 2001] [Lesser et al., 2003] [Mangrum and DiMarco, 2000] [Hui et al., 2017]. However, the prediction of bradycardia is yet to be explored.

3.7 Summary and Concluding Remarks

This chapter shows that physiological time series analysis has mainly been limited to classification based approaches with only single-step predictions. Furthermore, the techniques are mainly applied to a variety of different medical applications. Summaries of classification approaches can be seen in Tables 3.1 and 3.2. However, only a few works consider actual time series forecasting of physiological data. The few that do have not fully explored the range of available approaches and can be considered to posses a range of limitations. Moreover, potential real world applications were not that apparent. A summary of regression based approaches can be seen in Tables 3.3 and 3.4. Machine learning algorithms used for time series classification of physiological data have not fully considered the temporal character and long-term dependencies of time series. Existing work on time series forecasting of physiological data have considered machine learning algorithms such as LSTM and BILSTM to take account of the temporal nature of the vital signs. However, more recently it was found that

combinations of different machine learning algorithms perform better for applications such as pollution and power demand forecasting. Moreover, time series analysis in clinical event prediction have only considered time series classification which does not provide any further insight into an event. The focus of the thesis is to explore the scope of different machine learning techniques in the forecasting of physiological time series data and to apply the forecast outcome to help to intervene in dangerous clinical events. Forecast strategies and different data approaches are investigated in forecasting physiological time series data to help addresses the limitations of the existing literature. Furthermore, a hybrid regression model is also developed to help forecast the scenarios of clinical events, with results that are compared with the best classification based approach.

Chapter 4

Forecasting Strategies and Data Approaches for Machine Learning of Physiological Data

This chapter explores the scope of machine learning models along with different strategies to forecast patient physiological time series data 30 minutes in advance. Comparison of different forecasting strategies will help to reveal what strategies might be better for use in a forecast model for physiological time series data. A forecast horizon of 30 minutes is considered a long horizon in medical applications, especially in hypertensive event prediction [Lee and Mark, 2010b] [Ghosh et al., 2016]. Moreover, this chapter also examines two different approaches to forecasting mean arterial blood pressure (MAP). Univariate and multivariate forecasting approach are compared using a combination of machine learning models and forecasting strategies. Comparison between univariate and multivariate approach would explore whether an additional vital sign such as HR could improve the forecast accuracy of the response variable or perhaps only the past data of the response variable is good enough in forecasting. The univariate case is where a single variable is used as an input, and the single variable is predicted as an output. The multivariate case is where multivariable time series data are used as the input, and single variable data are predicted as the output. This chapter also compares single-step and multi-step forecasting by comparing MIMO and DIRMO strategies. Patients' physiological time series data (BP and HR) has been extracted

from the MIMIC II database and used for this comparative study. LSTM, BILSTM and CNN machine learning models were considered along with MIMO and DIRMO forecasting strategies. Forecast models were developed following both univariate and multivariate data. Blood pressure is of interest because forecasting blood pressure could potentially help a clinicians to take preventive steps to help avoid dangerous situations.

The experiments that will be conducted in this chapter are listed below:

- Comparison of different forecasting strategies to identify the best strategy for forecasting of physiological time series data.
- Consideration of LSTM RNN and BI-LSTM RNN algorithms with different forecasting strategies to build forecasting models for vital signs 30 minutes in advance.
- Comparison between LSTM RNN and BI-LSTM RNN algorithms in forecasting physiological time series data.
- Descriptive statistical analyses of physiological time series data and its effect on forecasting.
- Comparison of univariate and multivariate approaches in order to identify the best data approach in forecasting of BP time series data.
- Apllication of LSTM RNN, BILSTM RNN and CNN algorithm using the MIMO and DIRMO strategies to build forecast models to forecast BP 30 minutes in advance.
- Comparison between the MIMO and DIRMO forecast strategies in order to identify whether the single or multi-step forecast is favourable for forecasting blood pressure time series data.
- Forecast horizon analysis and their effects on forecast performance.

4.1 Methodology

The following section describes forecasting analyses on physiological datasets using the strategies described in Section 2.2. In this chapter, 12 forecasting models are

developed using python, by combining two machine learning algorithms, each using six forecast strategies. The performance of all the developed models are compared in the forecasting of heart rate and blood pressure using data from the MIMIC 2 database. This has been considered to find the best forecast strategy in physiological time series forecasting.

Following this, time series forecasting on physiological data sets are performed to find the best approach between univariate and multivariate data. Univariate and multivariate time series forecasting is compared using three different machine learning algorithms; LSTM, BILSTM and CNN, each implemented using the best two forecast strategies MIMO and DIRMO, as identified here in the chapter. The result of the combination is 12 forecast models using three machine learning algorithms and two forecast strategies, each implemented for both the univariate and multivariate cases. Forecast performance of all models in forecasting blood pressure is compared. The generic architecture of the models is presented in Fig. 4.1.

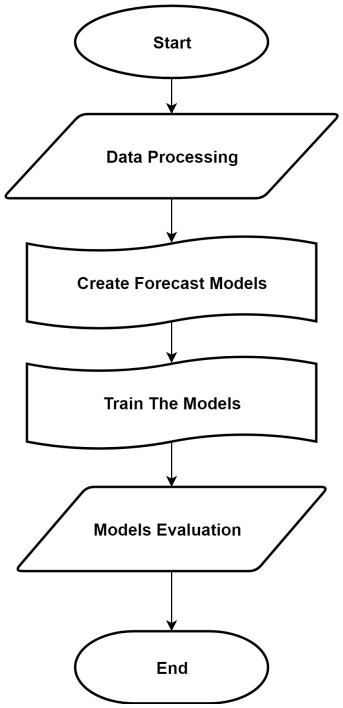


Figure 4.1: Model creation and evaluation procedure

Table 4.1: Overview of used datasets

Overview of Datasets	
No of Time Series Data	30 Patients'
Univariate data	BP
Multivariate data	HR & BP
Min Data Points of a Patients'	7904 min
Max Data Points of a Patients'	19669 min
Missing Values	No

4.2 Dataset

For the experiments in this chapter, minute by minute mean arterial Blood Pressure (BP) and Heart Rate (HR) time series data of 30 patients' have been extracted from the MIMIC 2 database. Thirty different sequences from 30 patients' are then predicted using the developed models and their average was considered to evaluate model performance. This scenario also somehow replicates a cross-validation scenario as 30 different forecast sequences are used to evaluate the performance rather a single forecast sequence. Experimental datasets were selected from the hypotension group of an ICD-9 code. The reason for choosing such groups is that it can be used for hypotensive and bradycardia event prediction, which is further investigated in chapter 5. During the data selection procedure here, it was ensured that there were no missing values on the selected time series. However, there are different techniques out there to impute the missing values but such scenarios are outside the scope of this work. The time series were re-scaled to values between -1 and 1 using MinMaxScaler function of scikit-learn. This is because the LSTM and BILSTM model requires data to be within the scale of the activation function (tanh) of the network [McKinney, 2012]. However, they were inverse transformed to the original values. Overview of the extracted time series data is presented in Table 4.1. The max and min data point of patients' are 19669 and 7904 respectively, whereas the other 28 patients' data points varies in between 19669 and 7904.

4.3 Data Preparation of Time Series

To undertake supervised learning, a time series dataset needs to be converted to a form that can be used in a supervised training process. The value of parameter Y (outcome) changes according to the strategy's output size in (Table 2.2), which is controlled by the parameter H.

Nevertheless, the underlying algorithm remains the same, which is as follows:

- Create a sliding window of size n_seq;
- Specify a parameter *step*;
- Slide the sliding window against the time series. The elements which are inside this sliding window are $\vec{X}$ of the supervised dataset;
- Element *steps* ahead of the last element in the window is the called the $\vec{Y}$ of the supervised dataset;
- Another parameter, H controls how many values correspond to each row of Y.

Parameters will change for each strategy in the sliding window technique. n_seq and *step* values are set to 30 and 1 respectively for all forecast strategies. This is because models will be fed with 30 minutes data as input (n_seq) to predict the output Y. However, in the sample, Y will vary according to the forecasting strategy with a parameter H. Overall, each sample will have two sections : input n_seq and output Y. This is explained below with a simple example. In the example, (1,2,3,4,5,6) is a univariate time series data, and it is required to forecast the last time point (6) from the time series. To do this, using supervised machine learning technique, the dataset will be processed into samples which will have input and output. The samples in the dataset will be used to train and test the models. Such supervised dataset can be created following an algorithm given below. In this example, following an iterative strategy, the parameters are set as $H = 1$, $n\_seq = 3$ and $step$=1, where H is the number of data points that are required to be predicted, *step* is the number of forecast steps, and n_seq is the past data prior to the predicted data points. Following these parameter values, the supervised dataset will look like:

- Sample 1: [1, 2, 3] [4]
- Sample 2: [2, 3, 4] [5]
- Sample 3: [3, 4, 5] [6]

The first two samples of the supervised dataset can be used to train a model to forecast the last point [6] of the time series. To forecast the last time point [6], the model will be fed with [3, 4, 5] as input and following this model will provide the forecast value of last time point of the time series. Different forecasting strategies can be implemented by varying the chosen parameters.

4.3.1 Dataset for Supervised Univariate Forecasting

BP time series data are the source of the univariate case. The univariate time series data set is converted samples each consisting of an observation window of 30 minutes of BP, and a target window of 30 minutes of BP for the MIMO strategy and 10 minutes of BP for the DIRMO strategy.

4.3.2 Dataset for Supervised Multivariate Forecasting

The source of the multivariate data is BP and HR time series data and are used to forecast future blood pressure. The multivariate time series data set is converted to samples each consisting of an observation window of 60 minutes, including 30 minutes of BP and HR each and a target window of 30 min of BP for the MIMO strategy and 10 min of BP for the DIRMO strategy.

4.4 Forecast Model Formulation

The LSTM and BILSTM RNNs are designed here with a network structure consisting of 1 hidden layer with 10 LSTM units. Then an output layer with a tanh activation and target window (Y) as output values which varies following the forecasting strategy. LSTM is stateful in the designed network, and the network was fitted with five epochs. The CNN was designed with a network structure consisting of one hidden convolutional layer followed by a max-pooling layer. The filter maps are then flattened before being

interpreted by a dense layer outputting a prediction. The output layer consists of a tanh activation. The output value was set equal to the target window (Y). The batch size of the networks was set to 1. The target window was varied according to the number of time steps over which a forecast was required. This also varies depending on the forecast strategy. The number of neurons in the output layer also differs depending on the forecasting strategy. The MIMO and DIRMO models predict multiple points, so more than one neuron is required at the output layer of the model. In MIMO, the number of neurons in the output layer is equal to the number of predictions needed in each regression. Whereas in DIRMO, the number of neurons in the output layer is calculated, dividing the number of prediction points by the number of models. The network also uses the RMSE as a loss function and the ADAM algorithm [Kingma and Ba, 2014] as an optimizer. However, the parameters of the developed models were not tuned. This is because the experiments in this chapter were not performed for a specific medical problem. Instead, the main aim of the experiments was to compare the univariate and multivariate approach considering different forecast strategies and machine learning algorithms. Parameters tuning is performed in chapter 5, where physiological time series forecasting was applied to specific medical applications.

All models were developed using the Python ecosystem [McKinney, 2012]. Keras deep learning library using the TensorFlow backend was used for the CNN, LSTM and BILSTM models [Chollet et al., 2015]. The developed forecast models aim to forecast physiological data 30 minutes in advance. To perform forecasting using the models, the datasets consisting of samples, were split into training and test sets. Samples were created from a patient physiological time series data following the methodology mentioned in Section 4.3. This process is applied to all 30 patients' considered here in the experiment to create 30 different dataset. Samples that consist of the last 30 data points of physiological times series data are used for testing in all models and the samples consisting of data points prior to that are used for training. Sample size depends on data points of the time series, input and output size of the sample. Moreover, the output of the sample depends on forecast strategies and input size depends on the user-defined input sequence. For example: to forecast a patients blood pressure 30 minutes in advance using the combination of univariate data and an iterative strategy, first the time series data of physiological data will be extracted. Supposing

the time-series data have 1000 data points. The sampling frequency of the data is a minute. Consider feeding 30 minutes of data as an input to the model to forecast the next 30 minutes. Then the input size of the sample would be 30 minutes and the output will be 30 minutes to forecast across the 30 minutes. However, as mentioned, that sample output depends on the forecast strategy. For this example, consider an iterative strategy. Following the description in subsection 2.2.1, the output size of the sample will be 1 as the model has a recursive strategy and forecasts a single point at a time. The input size of 30 and output size of 1 will produce 970 samples from 1000 data points. Where 941 to 970 samples consist of the last 30 minutes of the data points from the time series which are then used to make the prediction of the next 30 minutes. This forecast results are then used to help contribute towards the error metric by comparing with the last 30 data points (971-1000). Alternatively, the forecast result may then form an input to subsequent stages of processing. The remaining samples (1-940) are used for training and consist of data points prior to the 30 minutes sequence which are forecast. For a multivariate scenario, the data preparation follows the description in subsection 4.3.2. For the multivariate approach, the above example has the same number of samples and output shape but the input size of the samples will be 60. This is because an additional 30 data points of HR are combined with the 30 BP data points. The training data are used to train the models, and the test data are used for performance characterization. In the testing phase, the prediction function of the model is called to make predictions on given input values (X). In this chapter, 30 minutes ahead, forecasting was performed on physiological data, and the target value (Y) was set following different forecast strategies. To measure the performance of the models, the RMSE is calculated here using

$$\bar{e} = \sqrt{\frac{\sum_{i=1}^{m} (y_i - \widehat{y}_i)^2}{m}} \tag{4.1}$$

where, y_i are actual values, $\widehat{y}_i$ are forecast values and m is the number of target output data. The standard deviation of the RMSE was also calculated accordingly.

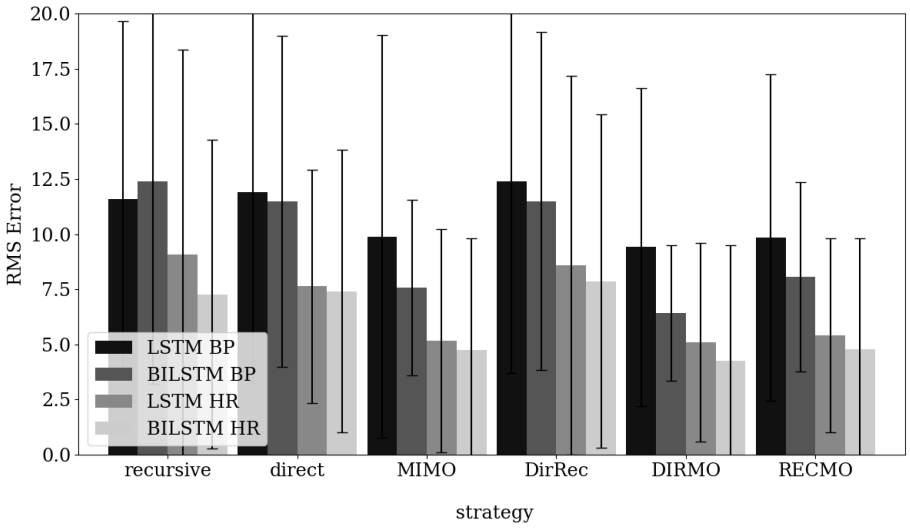


Figure 4.2: Performance of LSTM and BILSTM models with different forecast strategies in forecasting Blood Pressure (BP) and Heart Rate (HR) 30 minutes in advance.

4.5 Experimental Results From Comparison of Forecast Strategies

The performance of the 12 different forecasting models was assessed by forecasting HR and BP. The performance was evaluated based on the average RMS Error and standard deviation of thirty patients' and are plotted in Fig. 4.2. Details of the forecast models performance on BP using LSTM and BI-LSTM algorithm along with forecast strategies are available in Appendix C and D, respectively. Moreover, details of the forecast models performance on HR using LSTM and BI-LSTM algorithm along with forecast strategies are available in Appendix E and F, respectively.

Figs. 4.3 and 4.4 show the forecast performance of the LSTM model and BI-LSTM model on each patient along with different forecast strategies in forecasting BP 30 minutes in advance respectively. Figs. 4.5 and 4.6 show the forecast performance of the LSTM model and BI-LSTM model on each patient along with different forecast strategies in forecasting HR 30 minutes in advance respectively.

Table 4.2 and 4.3 consist of the descriptive statistics of the patients' BP and HR data, which were forecasted with high and low RMS Errors, respectively.

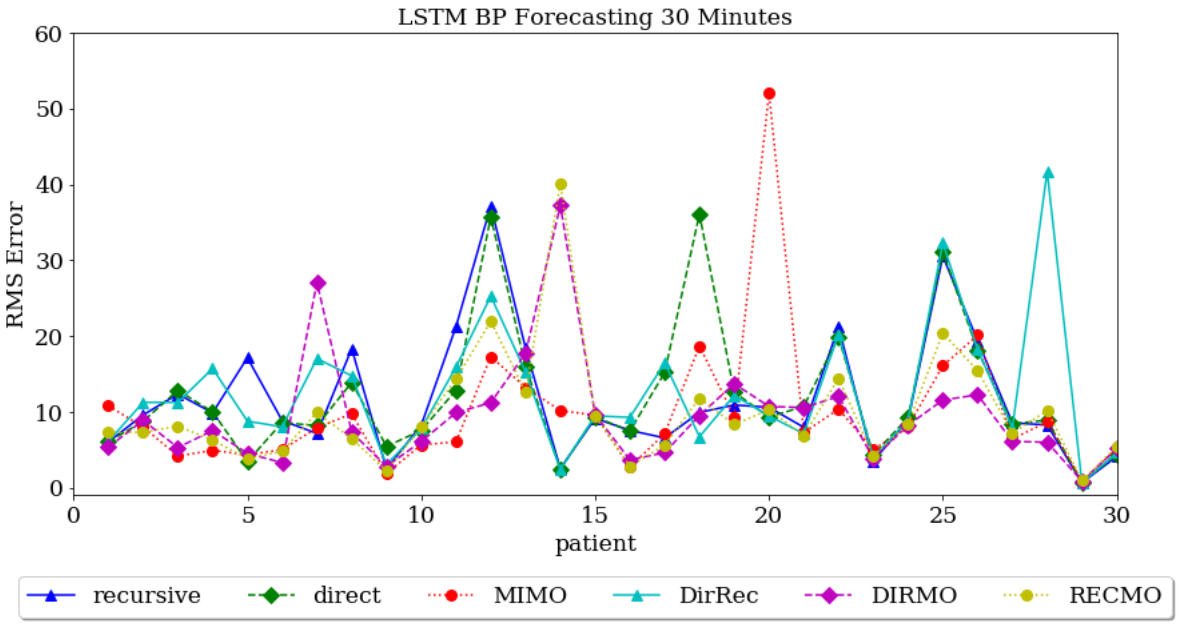


Figure 4.3: RMS errors per patient with LSTM models for BP forecasting for the different strategies

4.5.1 Comparison of Forecasting Strategy

Six different forecasting strategies have been used in the experiment. Average RMS error of each strategy is plotted in Figs. 4.2 and it can be observed that the MIMO, DIRMO and RECMO forecasting strategies appear to exhibit lower RMS errors, with DIRMO being the best overall. The MIMO and RECMO strategies were in close competition but both were unable to outperform the DIRMO strategy. Traditional strategies like the Recursive, Direct and DirRec forecast performance showed poor performance in forecasting BP and HR.

4.5.2 Comparison of LSTM and BILSTM

From Fig. 4.2, it can be observed that the BILSTM model formulations appear to be consistently better than the equivalent LSTM models. The improvement in the performance of the BILSTM models in comparison to the LSTM models is also confirmed by the overall mean performances, as shown in Table 4.4. This compares the overall performance between the LSTM and BILSTM models, calculated across all strategies

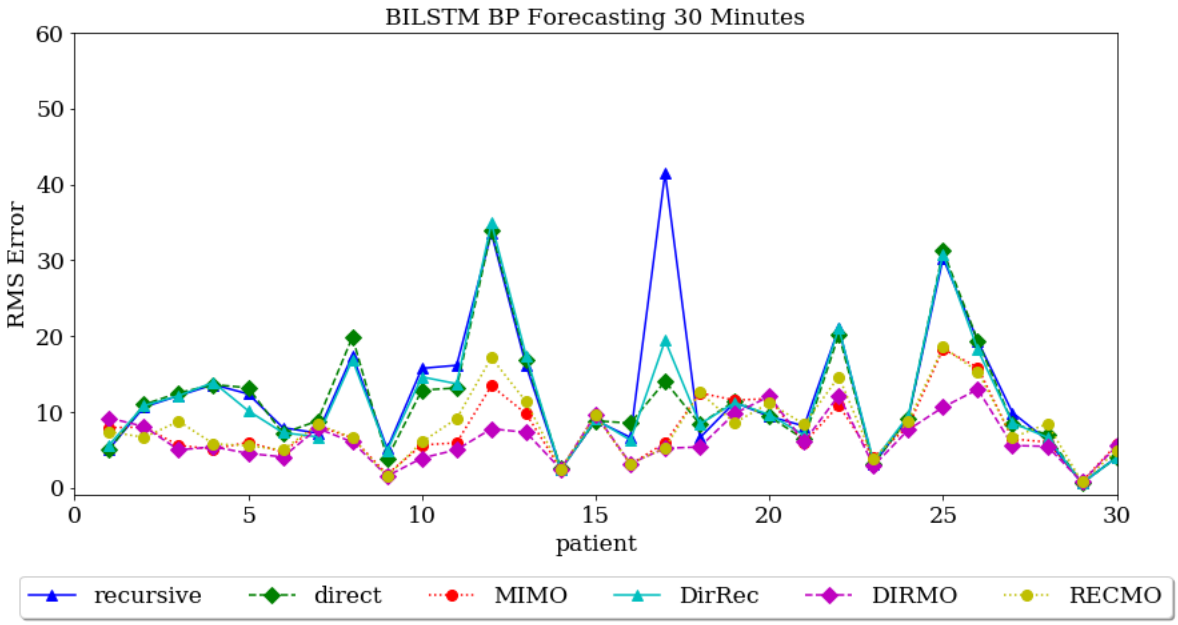


Figure 4.4: RMS errors per patient with BILSTM models for BP forecasting for the different strategies

for both models and both vital sign types (BP and HR).

4.5.3 Descriptive Statistics Analysis

A breakdown of the results per patient are given in Figs. 4.3, 4.4, 4.5 and 4.6. It can be seen that several sequences have a similar forecast performance. However, a few of the sequences for individual patients have considerably larger RMS errors. It is therefore interesting to compare the summary statistics for the sequences from different patients' exhibiting these differing errors.

Table 4.2 and 4.3 compares the counts, means, standard deviations and different percentiles for the sequences exhibiting either relatively large or relatively small RMS errors of BP and HR respectively. It appears to show that relatively more significant standard deviations are seen for the sequences that seemed to be more challenging to forecast. This is found applicable in both BP and HR forecasting.

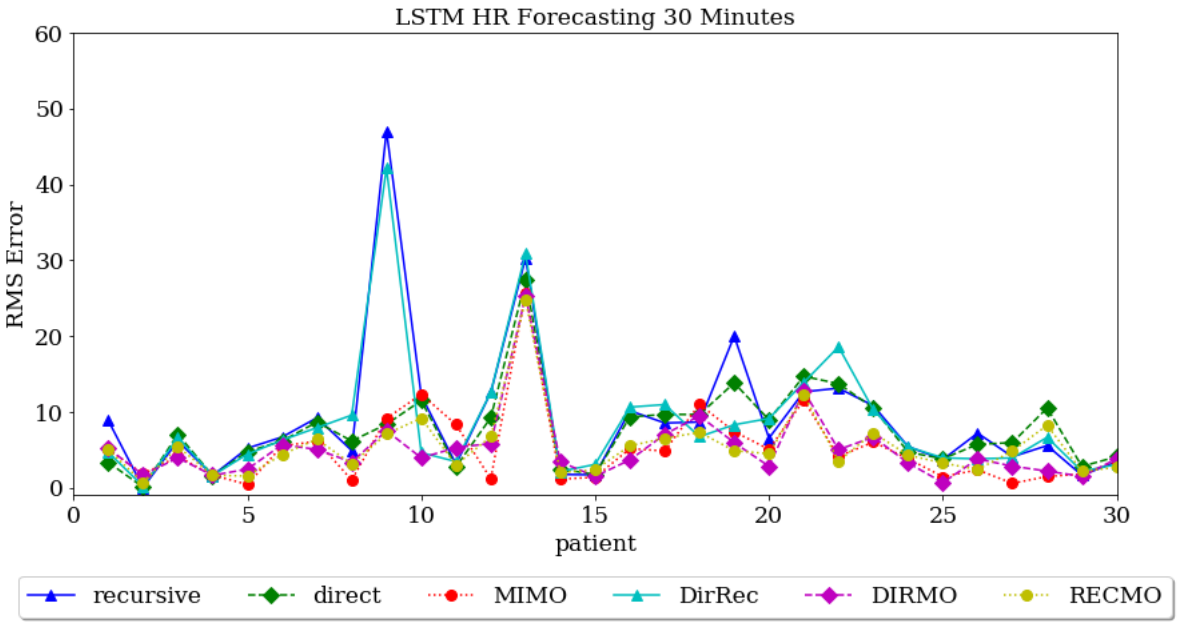


Figure 4.5: RMS errors per patient with LSTM models for HR forecasting for the different strategies

4.6 Experimental Results From Comparison of Data Approaches

Performance of the forecast models developed with the MIMO strategy based on 30 patients' in comparing univariate and multivariate approaches are shown in Figs. 4.7(a), 4.7(c) and 4.7(e). Performance of the forecast models developed with the DIRMO forecast strategy are shown in Figs. 4.7(b), 4.7(d) and 4.7(f). Average RMSEs and standard deviations of patients' for all models are shown in Fig. 4.9 in comparing the univariate and multivariate approaches. Performance of the forecast models for the univariate case in comparing single-step and multi-step forecast are shown in Figs. 4.8(a), 4.8(c) and 4.8(e). These can be compared with the performance of the forecast models using the multivariate approach which are shown in Figs. 4.8(b), 4.8(d) and 4.8(f). Average RMSE of patients' of univariate and multivariate approaches are shown in Fig. 4.10 comparing single-step and multi-step forecasting. Details of the performance of CNN, LSTM and BILSTM are available in Appendix G, H and I respectively.

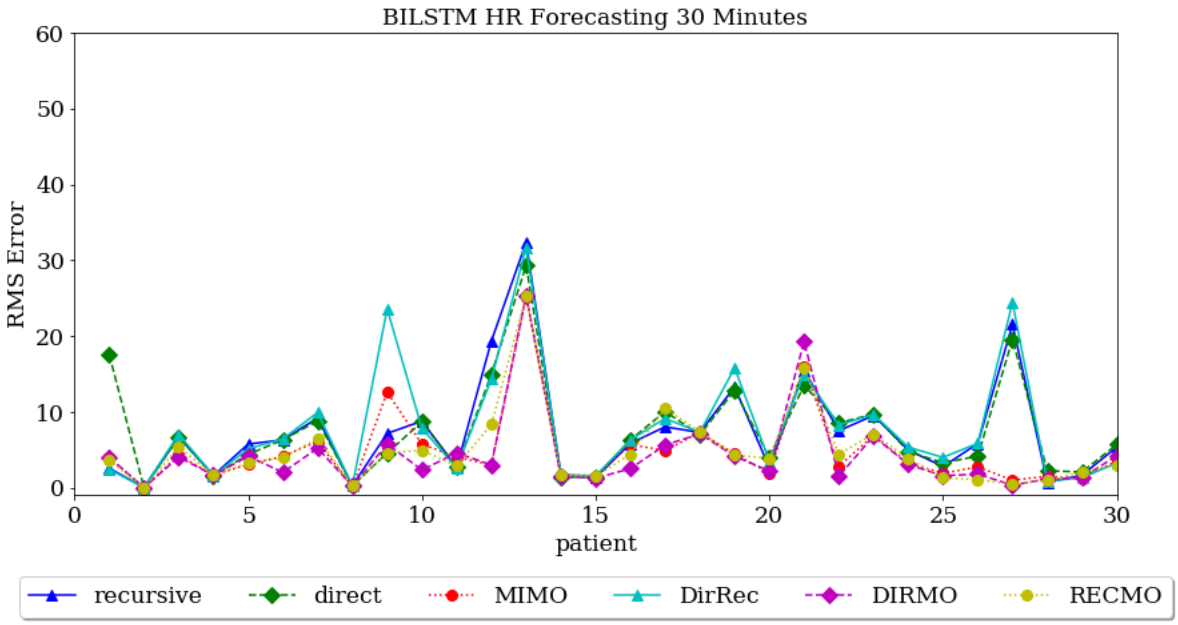


Figure 4.6: RMS errors per patient with BILSTM models for HR forecasting for the different strategies

4.6.1 Comparison of Univariate and Multivariate Approaches

Six forecast models have been tested combining three machine learning models and two forecast strategies, covering both univariate and multivariate techniques. The performance of the machine learning models using a single-step forecast strategy (MIMO) appear to show in Figs. 4.7(a), 4.7(c) and 4.7(e) that the multivariate approach provides better performance compared to the univariate approach. Similarly, for the multi-step (DIRMO) results, as shown in Figs. 4.7(b), 4.7(d) and 4.7(f). It also appears to be the case that the multivariate cases outperforms the univariate ones in all forecast models. To explore further, average RMSE and the RMSE standard deviation from all patients' are also taken into consideration and are shown in Fig. 4.9. Careful observation of the average RMSE and the RMSE standard deviations of all patients' appears to show that for both scenarios, the multivariate approach outperforms the univariate approach in all forecast models. So overall, it appears that multivariate techniques can provide better performance in forecasting blood pressure 30 minutes in advance compared to univariate techniques.

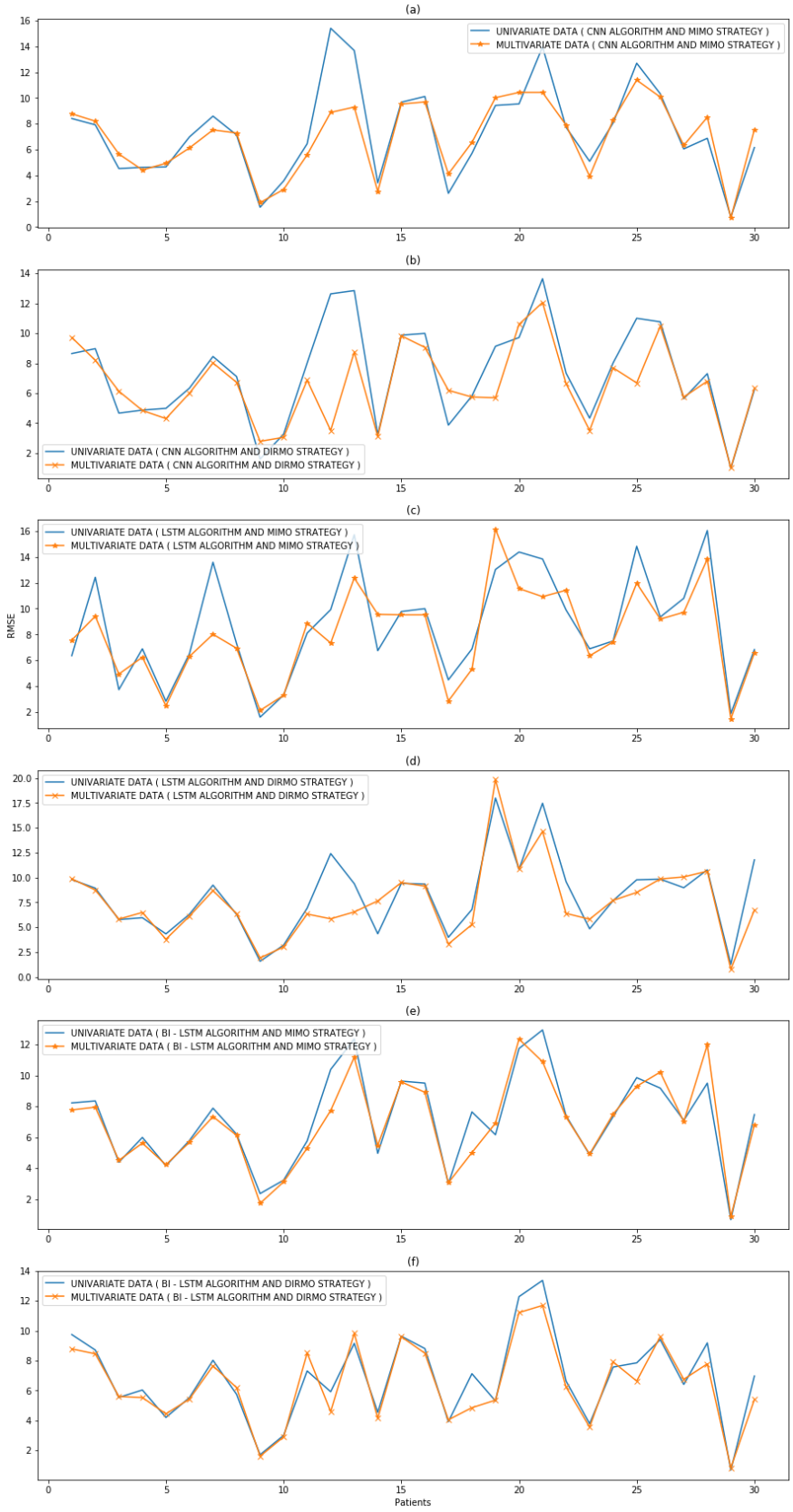


Figure 4.7: Mean RMSE comparisons of univariate and multivariate cases considering the performance of each of the machine learning models and forecast strategies for each of the 30 patients'

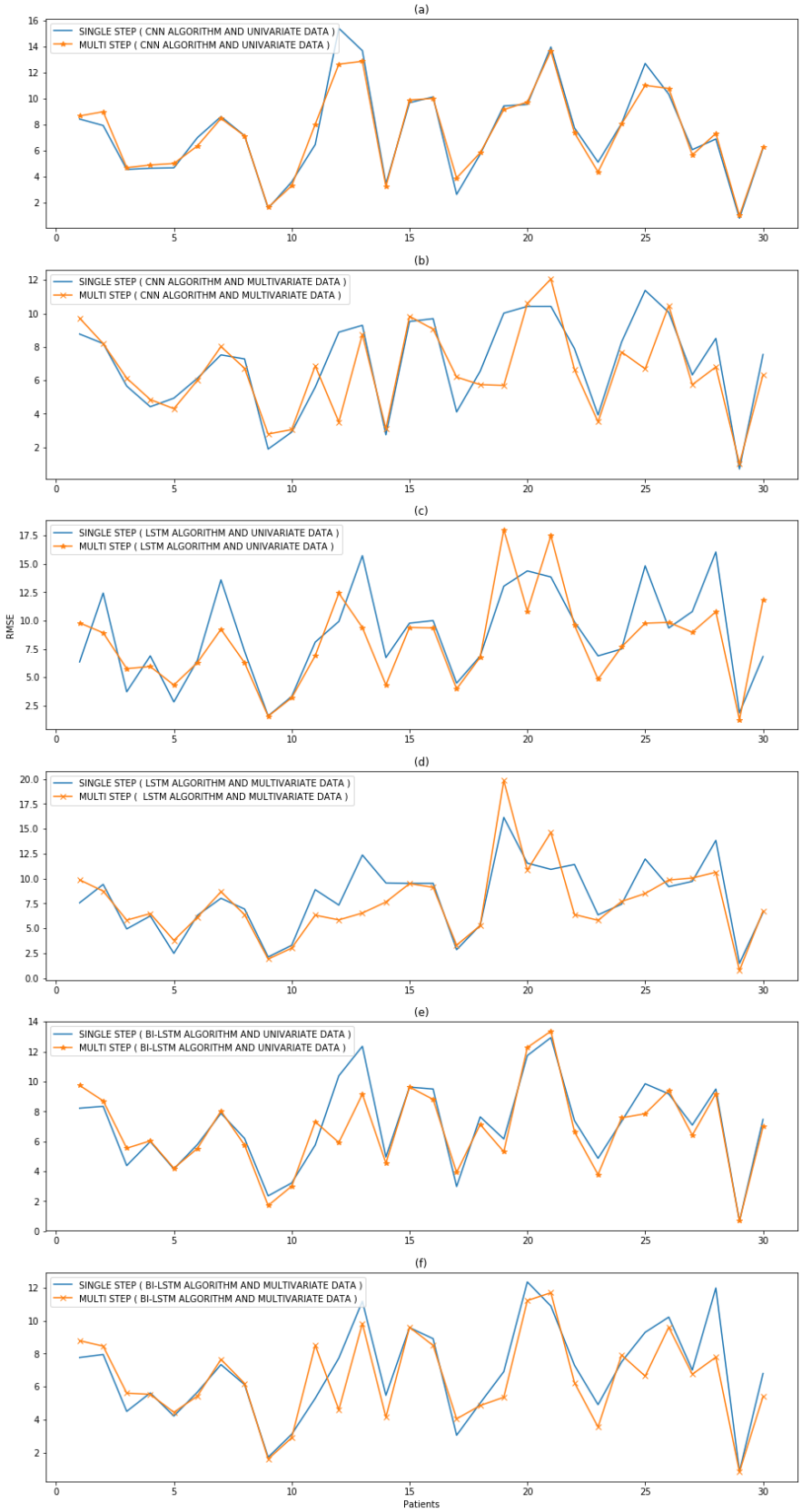


Figure 4.8: Mean RMSE comparisons of single-step and multi-step forecasting considering the performance of each of the machine learning models for the univariate and multivariate approaches for each of the 30 patients'

Table 4.2: Descriptive statistics of the patients' with low and high RMS errors for BP forecasting.

	Patients' with high RMSE in BP forecasting				
	11	12	14	17	25
COUNT	30759	19669	11316	8443	8480
MEAN	92.06	74.87	86.42	77.71	91.91
STD	19.98	10.98	36.01	11.86	15.88
25%	79.90	68.80	68.40	71.20	81.30
50%	92.30	74.60	77.00	78.30	92.15
75%	104.20	80.30	88.90	83.90	103.10
MAX	357.70	284.70	351.00	283.90	292.10
	Patients' with low RMSE in BP forecasting				
	9	23	29		
COUNT	10086	8125	8797		
MEAN	68.084	76.35	69.30		
STD	8.15	14.07	8.84		
25%	63.70	68.30	64.30		
50%	67.50	74.50	68.90		
75%	71.70	81.60	73.70		
MAX	215.50	280.70	215.50		

4.6.2 Comparison of Single-step and Multi-step Forecasting

The performance of single-step and multi-step forecasting of machine learning models for univariate data are plotted in Figs. 4.8(a), 4.8(c) and 4.8(e). It can be seen that multi-step forecasting (DIRMO) appears to perform better in comparison to single-step forecasting (MIMO). A similar observation can be made on Figs. 4.8(b), 4.8(d) and 4.8(f) for the multivariate data. Average RMSE and the RMSE standard deviation of 30 patients' blood pressure forecasting were also taken into consideration for machine learning models with both single-step and multi-step approaches with the outcomes shown in Fig. 4.10. These show that multi-step forecasting provides lower RMSE and standard deviation in comparison to single-step forecasting in all scenario. Thus, it

Table 4.3: Descriptive statistics of the patients' with low and high RMS errors for HR forecasting.

	Patients' with high RMSE in HR forecasting				
	9	13	21	27	
COUNT	10086	7904	6831	10692	
MEAN	80.16	82.98	93.14	82.83	
STD	16.18	10.56	14.11	12.62	
25%	67.00	75.90	81.65	73.00	
50%	76.00	81.50	92.50	81.00	
75%	91.00	89.30	104.30	93.00	
MAX	139.80	131.20	167.40	148.00	
	Patients' with low RMSE in HR forecasting				
	20	29	30		
COUNT	9051	8797	7402		
MEAN	90.50	87.70	90.56		
STD	10.97	9.69	10.48		
25%	82.40	80.50	82.72		
50%	87.90	88.30	91.20		
75%	97.60	94.20	98.60		
MAX	171.10	118.00	121.20		

Table 4.4: Comparison of LSTM and BILSTM overall errors and standard deviations. Here, the average of the RMS errors is calculated across the different strategies to enable a simpler comparison between LSTM and BILSTM.

	BP		HR	
	LSTM	BILSTM	LSTM	BILSTM
mean	11	10	7	6
st.dev.	8	6	6	6

can be tentatively concluded that overall, multi-step forecasting performs better than single-step forecasting in forecasting blood pressure 30 minutes in advance.

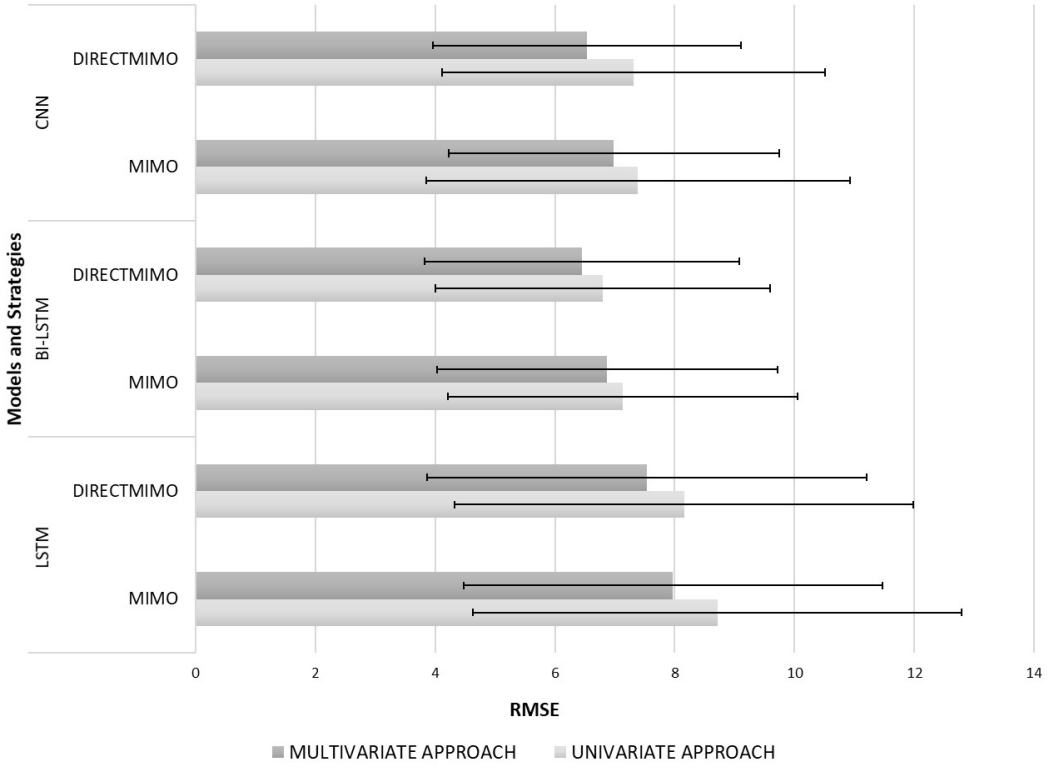


Figure 4.9: Comparison of univariate and multivariate cases considering the average RMS Error and standard deviation of all patients'

4.6.3 Best Forecasting Model

Considering the empirical results shown here, it can be tentatively concluded that a BILSTM forecasting model along with a DIRMO strategy and for the multivariate case seems to be the best model. From Fig. 4.9 , it is observable that BILSTM outperforms CNN and LSTM. From Fig. 4.9 it is also observable that the BILSTM forecasting model along with a DIRMO strategy and multivariate configurations show the best performance. Moreover, in Fig. 4.10, it appears that the BILSTM model outperforms the CNN and LSTM models for both the univariate and multivariate approaches.

4.6.4 Error Analysis in Different Forecast Horizons

So far, only a single forecast horizon has been considered for all forecast models. This has involved forecasting of BP, which was 30 minutes. However, a further range of

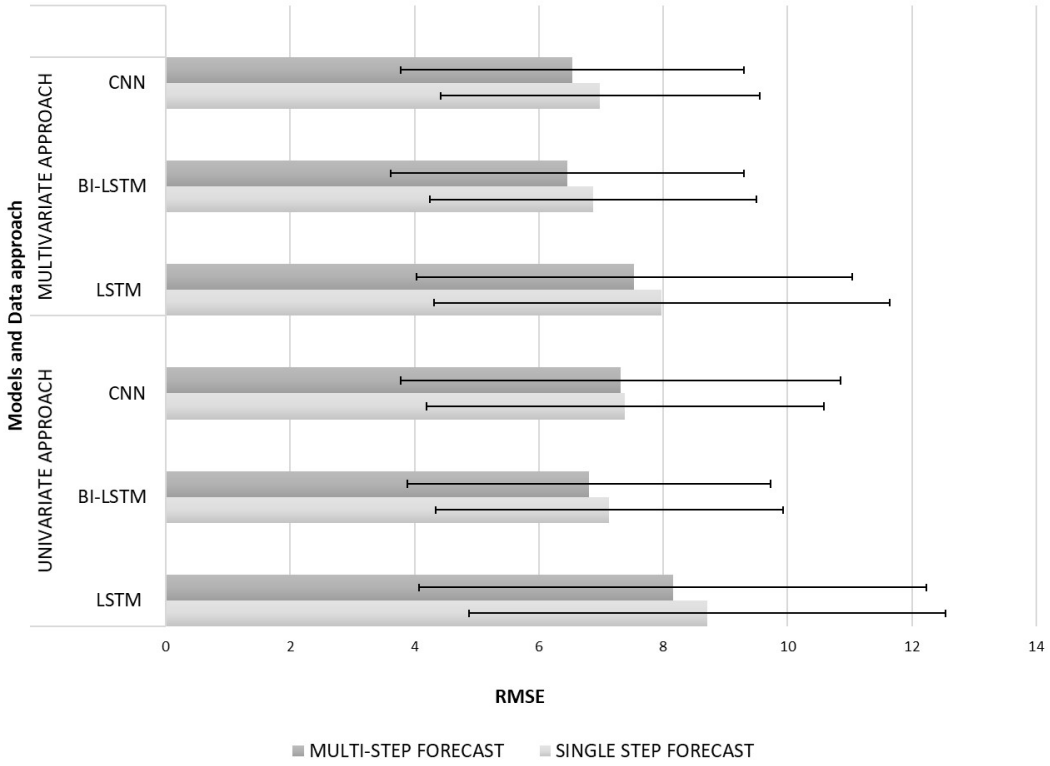


Figure 4.10: Comparison of single-step and multi-step forecasting considering the average RMS Error and standard deviation of all patients'

forecast horizons was also considered. The forecast horizon of up to 2 hours with an interval of 10 minutes is also considered. The forecasting error across this range is shown in Fig. 4.11 and Fig. 4.12 for the best forecast model (BILSTM, DIRMO, multivariate). Details of the error analysis in different forecast horizons are available in Appendix J. It is observable that as the forecast horizon increases, the RMSE increases. Furthermore, from 60 to 120 minutes, the RMSE increases linearly for almost all cases. In a few cases, the RMSE can be seen to be higher. This could be because of the inherent variability of these physiological sources of time series.

4.7 Discussion

Analysis of different forecast strategies can help in the selection of the best forecast strategy. This is important when building a forecast model for a particular application

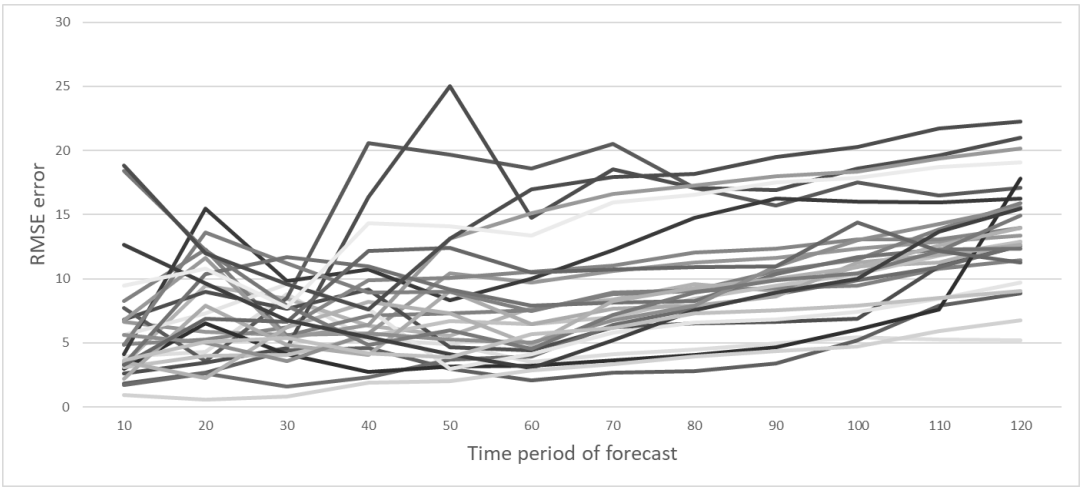


Figure 4.11: RMSE of all patients' blood pressure forecasting for different forecast horizons

such as for forecasting of physiological data. The results appear to suggest the best performing strategy is DIRMO. This appears to be in agreement with the work of Taieb et al. [Taieb et al., 2012]. There, various different types of MIMO and DIRMO were compared along with DIR, REC and DIRREC forecasting strategies. There DIRMO was also found to outperform MIMO but not in all cases: only when no input selection had taken place which is also the case here. Application of a multivariate machine learning model in forecasting BP allows the inter-relationship between BP and HR to be somewhat investigated, by implicitly modelling the correlation within the time series. This has resulted in improved accuracies when forecasting BP. This claim is based on observations for all six forecast models where three different machine learning models and two forecast strategies are combined. The multivariate approach, along with the BILSTM model and DIRMO forecast strategy, has proved to be the best model in forecasting BP 30 minutes in advance. Such modelling might be useful in potentially helping healthcare professionals to plan and make decisions. The multivariate approach provides insight into the dynamic relationships of the used variables, but in such cases, more variables, data points and datasets are required. Data points of all variables need to be measured at the same time period (uneven data recording), and the availability of such long datasets are infrequent and hard to extract. Moreover, Fig. 4.9 shows that the difference between univariate and multivariate approaches is not huge. However,

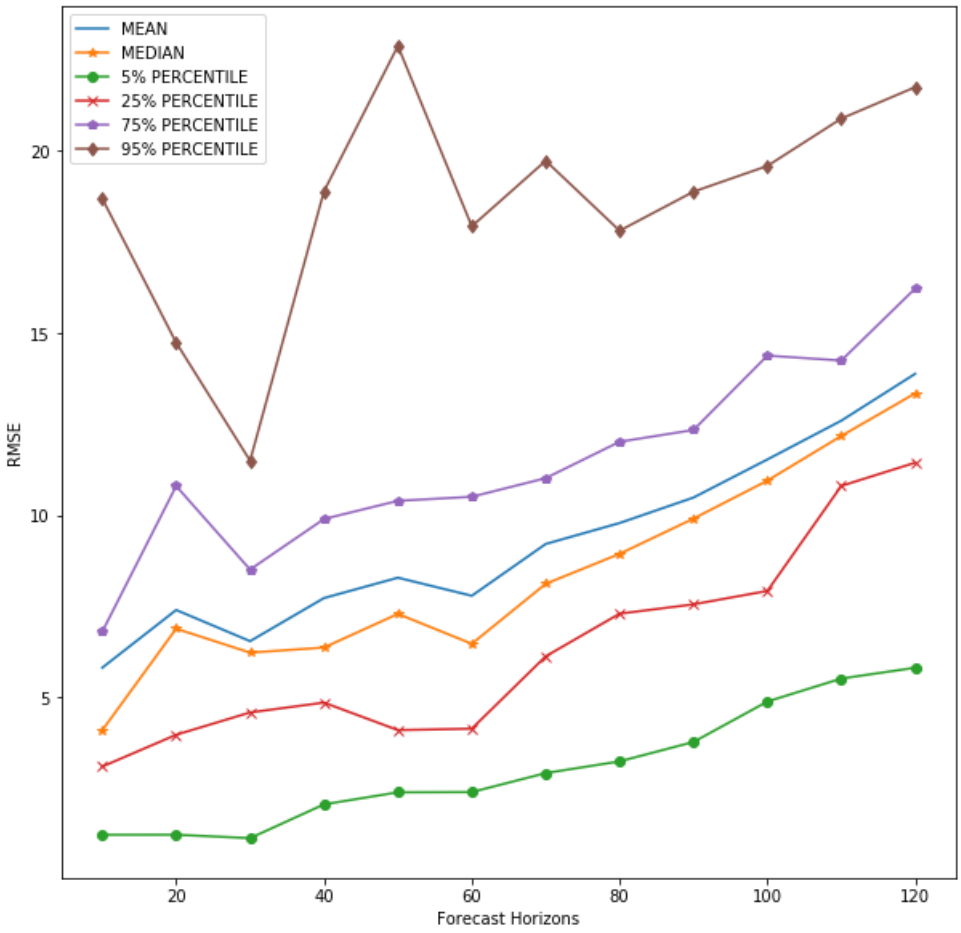


Figure 4.12: RMSE of patients' blood pressure forecasting for different forecast horizons

the multivariate approach does have the edge over the univariate approach. Thus when building a forecast model for a specific medical application, a trade-off needs to be considered between the availability of multivariate data compared to the univariate data. Moreover, the same observation can be made from the results in Fig. 4.10 . This shows the difference between single-step and multi-step approaches is minimal. Thus a trade-off between single and multi-step should be considered taking into account the complexity in fitting these two approaches in the forecast model. Single-step with MIMO is quite straight forward to fit in a forecast model with less complexity in data preparation and lower run time. In contrast to this, for multi-step, the data preparation is complex and the model runtime is higher than single-step.

4.8 Summary and Concluding Remarks

The results of this study appear to show that it is possible to forecast future values of physiological time series with relatively small error 30 minutes in advance. The comparative analysis between the forecast strategies revealed that the DIRMO and MIMO forecast strategies provide the best accuracy in forecasting BP and HR. The forecast models developed with MIMO, DIRMO and RECMO strategies provided predictive forecasts with low RMSE. This possibly suggests that they might have a future role to play in a system or systems, with further testing, to help in automated decision-making processes. The results of this study also appear to show that the BP forecast is further enhanced with the aid of HR information. The accuracy and the resulting forecast values are influenced by the length of time over which the future forecast is made (the horizon). The results shown here appear to demonstrate that multivariate time series modelling is more reliable in forecasting BP 30 minutes in advance. The multivariate approach, along with a BILSTM model and DIRMO strategy provide more accurate forecasting than the univariate approaches. This is also true for other machine learning models and forecast strategies for the somewhat limited data included here. It is also observed that the BILSTM model with a DIRMO strategy provides the lowest standard deviation, which makes this combination favourable for forecasting blood pressure to help healthcare professionals in decision making.

Chapter 5

Machine Learning of Times Series for Clinical Event Intervention

Hypotension is a problem with low blood pressure. It is seen to be a major risk factor of death in hospital intensive care units (see e.g. [Ghosh et al., 2016, Moody and Lehman, 2009, Lee and Mark, 2010a, Lee and Mark, 2010b]). Bradycardia is a low pulse rate. It is not usually considered a major risk factor of deaths, but it can be, particularly if a patient is ill and being treated with medicines that might adversely affect the pulse rate (see e.g. [Yong et al., 2015, Cheung et al., 2015, Kusumoto et al., 2018]). If the blood pressure is too low for a period of time then complications can arise, resulting in whole organ failure. Bradycardia is also a cause for concern. It can cause various cardiovascular problems including heart failure, sudden cardiac arrest, stroke and mortality. It is therefore of interest to be able to predict the possibility of hypotension and bradycardia and hopefully enable medical intervention to take place.

This chapter explores the scope of supervised machine learning in this predictive role. Hypotensive and bradycardia events can be detected by a medical expert from mean arterial blood pressure and heart rate time series data. Therefore the work of previous chapters is developed further here to simultaneously perform forecast prediction along with a further step of detection. The prediction models from previous chapters are also further developed here, again in a regression like role, to inform a human observer and is used to perform a binary detection step. A gap window is also given, to increase the amount of time that a medical expert could be given to warn of

the potential of a dangerous event.

MAP and heart rate times series data have been extracted from the MIMIC III database [Johnson et al., 2016b]. These data are converted to a supervised time series dataset through appropriate data processing. The samples consist of an observation window and the target window. These were used to predict the occurrence of events in a future time window. This is following a user-defined gap interval. Results suggest that the prediction performance decreases as the gap interval increases. This is true for both a purely classification-based approaches and when using the regression algorithm. Results appear to show that the regression model performs better than the classification model in predicting the events. This is when the forecast includes a gap window of up to 30 minutes. Moreover, the outcome of the forecast model with regression carries extra information for the user. This could be handy in decision making as it predicts continuous outcomes rather than a purely categorical outcome. Overall, the results appear to show that supervised machine learning can be a useful technique in predicting events. The reported results could be of interest to biomedical engineers, decision support system developers and healthcare professionals.

The contributions of this chapter are listed below:

- An application of a hybrid regression algorithm in intervening in critical events such as hypotensive and bradycardia events;
- A hybrid machine learning forecast model with a long forecast horizon and low error, to predict critical events;
- A data preparation technique which helps to improve the forecast model accuracy in event prediction;
- Models were validated on a dataset consisting of a large number of samples extracted from MIMIC-III [Johnson et al., 2016b]; a large-scale deidentified database.

5.1 Methodology

A Multiple Input and Multiple Output (MIMO) forecast strategy is used here for forecasting following the findings in chapter 4. Along with DIRMO (a combination of

Direct and MIMO strategies), these were found to provide the best performance on forecasting of physiological time series data. MIMO strategy forecasts a sequence of multiple outputs simultaneously from a single model M, i.e.(see e.g.(2.1.2))

$$\begin{aligned}\Big(F(t+H), F(t+H-1), ..., F(t+1)\Big) \\ = M(o(t), o(t-1), ..., o(t-n))\end{aligned} \tag{5.1}$$

where $o(t-1)$ is an observation at time $t-1$ and $F(t+H)$ is a forecast for time $t+H$. This means that the forecast model is using $n+1$ observations. Often, for applications such as time series forecasting of physiological data, a gap in time (window) Δt is excluded so that Δt is also included in the formulation, i.e.

$$\begin{aligned}\Big(F(t+H+\Delta t), F(t+H-1+\Delta t), ..., F(t+1+\Delta t)\Big) \\ = M(o(t), o(t-1), ..., o(t-n))\end{aligned} \tag{5.2}$$

This means the forecast is for Δt time points in the future. In contrast to this, we propose to forecast Δt time points in the future but also to include the gap window in the forecast, i.e.

$$\begin{aligned}\Big(&F(t+H+\Delta t), F(t+H-1+\Delta t), ..., \\ &F(t+1+\Delta t), F(t+\Delta t), F(t+\Delta t-1), ..., F(t+1)\Big) \\ &= M(o(t-1), o(t-2), ..., o(t-n)).\end{aligned} \tag{5.3}$$

A binary decision $D \in \{0, 1\}$ can be made regarding the presence of a clinical event. It can be made by first considering the forecast values over the period of time $t+\Delta t$ to $t+H+\Delta t$,

$$\Psi(t) = \Big\{\tau \Big| F(t+\tau) < \iota, \Delta t \leq \tau \leq \Delta t + H\Big\} \tag{5.4}$$

so that a decision can be made depending on the proportion p of forecast values less than the threshold ι. This can be expressed as follows

$$D(t) = \begin{cases} 1 & \text{if} \quad |\Psi(t)|/H > p; \\ 0 & \text{otherwise,} \end{cases} \tag{5.5}$$

where $|\Psi(t)|$ represents the cardinality of $\Psi(t)$, i.e. the number of forecast time points that are predicted to be below the threshold. The thresholds for the MAP and the heart rate are both taken to be 60.

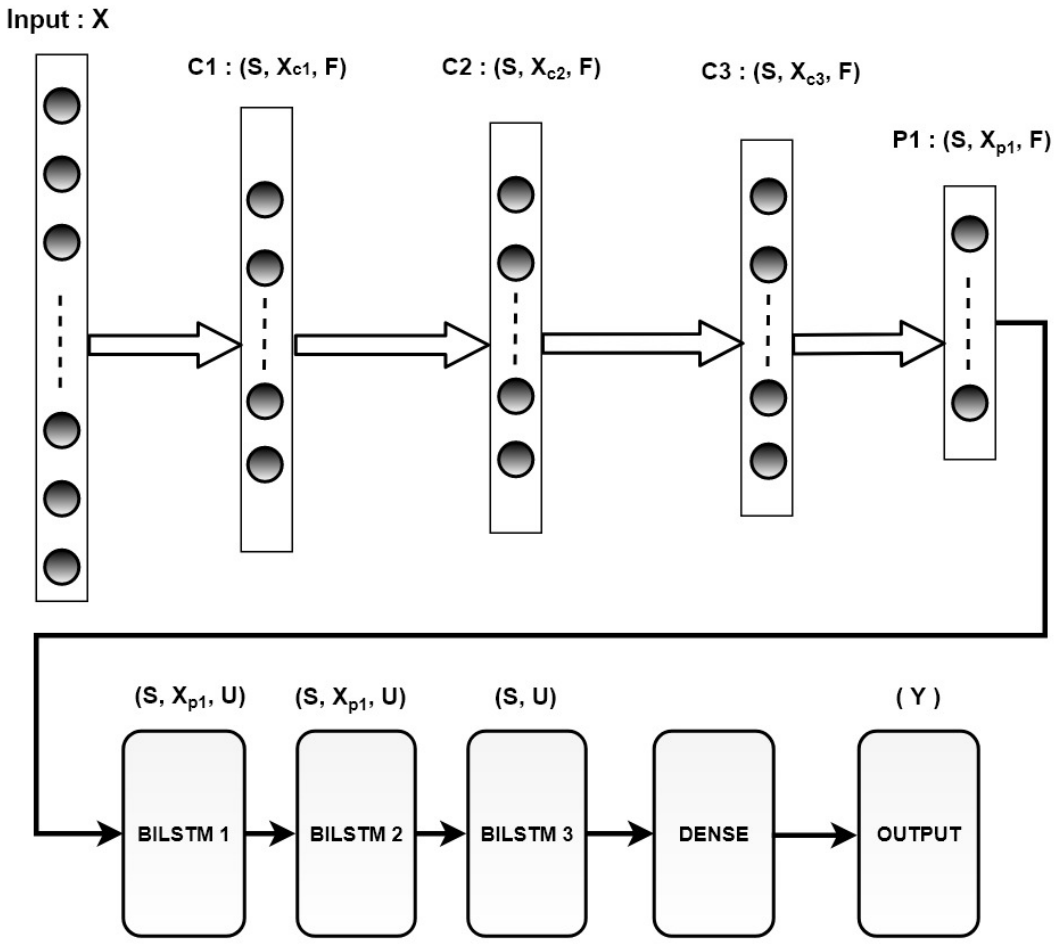


Figure 5.1: Regression based forecasting model with the hybrid CNN-BILSTM architecture. The output of this architecture can be considered to form the time series forecast $F(t+1)$,$F(t+2)$,...,$F(t+H+\Delta t)$. Here, X is the input, Cn is the convolutional layers, $P1$ is the polling layer, S is the number of samples, X_cn is the number of the compressed time steps, Xp_1 is the number of downsampled time steps, F is the number of filters, U is the number of units and Y is the output.

The time series regression model M as used above is implemented with the use of a machine learning architecture as shown in Fig. 5.1. It is based on a hybrid combination of CNN architecture layers and BI-LSTMs. This enables the benefits of both technologies to be simultaneously utilized. The network structure consists of three 1D convolutional layers, followed by a max-pooling layer. Convolutional layer merges and combines information from neighbouring time points. Each convolutional layer does this with the following mathematical operation to produce a feature map.

$$X(t) = \sum_{\tau=-[F/2]}^{+[F/2]} O(t)C(t-\tau) \tag{5.6}$$

where, $O(t)$ is an observation at time t, $C(t)$ is the filter kernel and F the size of the kernel. Convolutional operation slides the filter over the input data. Filter size was 3 for both hypotensive and bradycardia scenario. Stride specifies how to move the filter over the input at each time step. The stride value was set to one for both scenarios.

The feature map has a reduced size depending on filter size and stride values. However, to maintain the same dimensionality of the input, zero padding is used. After three convolutional layers, the resulting feature map is then fed into the pooling layer. The pooling layer reduces the dimensionality of the feature map. This is effective for faster training and to overcome overfitting issues. A max pooling operation is used with the same size as the pooling size and also the stride value of 2 in both cases. Such parameters halve the size of the feature map. Here the polling layer ($P1$) downsamples the feature map while keeping the important information. The convolutional process is repeated for all convolutional layers $C1$, $C2$ and $C3$ as shown in Fig. 5.1. The output shape of each convolutional layer is represented as (S, Xc, F). Here S represents the number of samples, Xc represents the number of the compressed time steps, and F represents the number of filters. The output shape of the pooling layer is represented with (S, Xp_1, F) as shown in Fig. 5.1 where S also represents the number of samples, Xp_1 represents the number of downsampled time steps and F represents the filter size. The value of Xp_1 is determined by the polling size and stride values. The output of the pooling layer ($P1$) is fed into a BILSTM layer ($BILSTM1$). Three BILSTM layers are used in the model architecture. The three-layers contribute to a deepening of the model BILSTM sub architecture [Graves et al., 2013]. Multiple BILSTM layers allow

the hidden state at each level to operate at a different time scale. Such layers help to create a more complex feature representation of the current input. A BILSTM layer outputs a sequence in the form of a vector which is used as an input to a subsequent BILSTM layer. The output shape of a BILSTM layer is represented as (S, Xp_1, U) as shown in Fig. 5.1. The number of samples is represented by S; Xp_1 represents the number of time steps in the output, and U represents the number of units. This hierarchy of hidden layers enables a more complex representation of time series data by capturing important information at different time scales. The output shape of the final BILSTM layer ($BILSTM3$) is then fed into a dense layer. The output of the dense layer is the output of the model and depends on a variable referred to as the 'unit value'. The unit parameter normally controls the size of the weight matrix and the bias vector in the dense layer. The size of the bias vector is given by the unit value. The unit's parameters, and the weight matrix depend on the size of the input data. However, the dot product produces an output that is the same size as the unit parameter.

A 3-layer architecture has been used by e.g. Funahashi's [Funahashi, 1989]. They proved that such an architecture, with a sigmoid activation function, can approximate any continuous input-output relation. However, more recent work (see e.g. [Liang and Srikant, 2016]) has shown that greater than three layers are needed to more efficiently approximate more complex higher dimensional functions. We utilize here a ten layer hybrid architecture consisting of CNN layers and BILSTM layers as shown in Fig. 5.1. Hybrid neural network models combine the advantages of CNNs and LSTMs. They have strengths and limitations in modelling [Pascanu et al., 2013a] [Mohamed et al., 2012]. It is believed that forecast performance can be improved by combining these networks in a unified framework. CNNs can be trained to recognize and to automatically help to extract salient features from an input feature vector. This helps to produce a more compact and efficient representation. The input feature vector can be from static data, or as done here, from sequential time series data. This then aids the BILSTM to identify the sequential dependencies in various samples of the data. Moreover, the choice of combining these algorithms is motivated by the work of Huang et al., which found for another application on forecasting of particulate matter in smart cities [Huang and Kuo, 2018] that a CNN-LSTM hybrid model performs better than

Table 5.1: CNN-BILSTM tuning parameters and ranges

Parameters	Range
filters	choice [2,3,4,5,6]
kernel_size	choice [3,4,5,6,7]
LSTM cell	choice [50, 61, 80, 95, 120]
dropout	uniform (0, 1)
recurrent_dropout	uniform (0, 1)
batch_size	choice [16,32,64,128,256,512]

CNNs and LSTMs individually.

The values of the hyperparameters of the model architecture summarised in Table 5.1 are determined with a tuning algorithm. The tuning algorithm conducts a randomized search over the given parameters. Each setting is sampled from a distribution over a range of possible parameter values [Bergstra and Bengio, 2012] [1]. The hyperparameter tuning process consists of a randomized search over a selection of ranges of the hyperparameter values as shown in Table 5.1. The randomized search seeks to minimize an error term. The mean square error is used here. Following the hyperparameter tuning, the best accuracy for a 30 minute gap window results in the following:

- filters: 3
- kernel_size: 6
- LSTM cell: 95
- dropout: .233
- recurrent_dropout: .913
- batch_size: 256

For tuning in each epoch, the model is trained with the best hyper tuned parameters found previously. For tuning the epoch, all the samples of the dataset are used. Epochs

[1] The tuning algorithm was created using a hyperparameter optimization package called 'Hyperopt', where 'hyperas' was used as a wrapper [Bergstra et al., 2013].

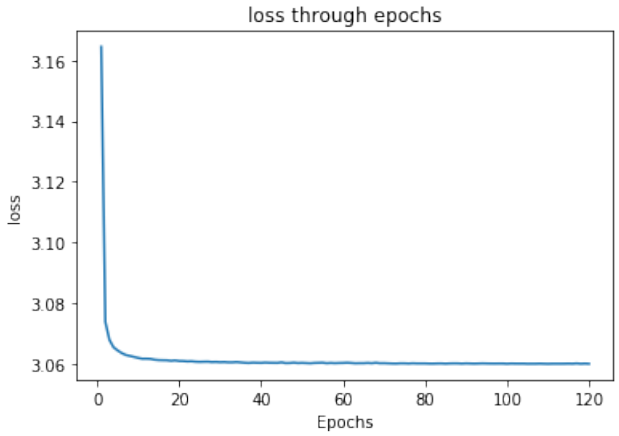


Figure 5.2: Training loss as a function epochs.

were set from 1 to 120 and training loss was recorded as shown in Fig. 5.2. This suggests that after 60 epochs, model training loss starts to converge. Following this, an epoch value of 60 is considered for the model.

All these tuned parameters were then used in the combined CNN-BILSTM forecast model[2]. The datasets were split into training and testing set. A set of 1000 samples of the datasets were used for testing. The rest of the samples were used for training, which are 20472 and 15702 respectively for hypotensive and bradycardia event. The training data were used to train the models, and the test data were used for performance characterization. This procedure was iterated ten times randomly, and the average was taken into consideration in performance characterization, which replicates cross validation scenario. The RMSE, MAE and MAPE techniques were used to measure the performance of the models. The actual outcome and the predicted outcome of the regression model is examined following the threshold value of 60. In predicting HE, if the regression model can correctly forecast 90% of the target window below the threshold, then the model predicts a hypotensive event accurately or can be classified as 1. It is vice versa for normotensive events and can be classified as 0. This allows calculating the sensitivity, specificity, precision and accuracy from the outcome of the model with the regression algorithm.

To forecast bradycardia, the hybrid CNN-BILSTM model is also used. Moreover, a

[2]Developed using Python ecosystem [McKinney, 2012].

different set of hyperparameter tuning was therefore required. The tuning methodology described above was followed for the CNN-BILSTM model for forecasting bradycardia. Different tuning parameters and their distributions were chosen, as shown in Table 5.1. Following hyperparameter tuning the parameters found for best accuracy for a 30 minutes gap window are:

- filters: 3
- kernel_size: 3
- LSTM cell: 80
- dropout: .209
- recurrent_dropout: .719
- batch_size: 128

These parameters were then used in the CNN-BILSTM model to forecast bradycardia.

5.2 Experimental Methodology

5.2.1 Data Extraction and Preparation

Mean Arterial blood Pressure (MAP) is used for the detection of hypotensive events. Therefore it is forecasted here to enable the prediction of hypotensive events. Thus, minute-by-minute time series of MAP were extracted from the matched subset of the MIMIC III database. The MAP is a measure of blood pressure which is calculated from systolic and diastolic pressure, see (3.1) [Mann et al., 2014].

Extracted MAP time series data has been used to create samples for the supervised dataset. Each sample from the supervised data consisted of three-time intervals. A graphical representation is shown in Fig. 5.3 (a) and described below:

- Observation window: a 120 minute observation window is considered here for the experiments. This can also be called the input feature vector of the prediction

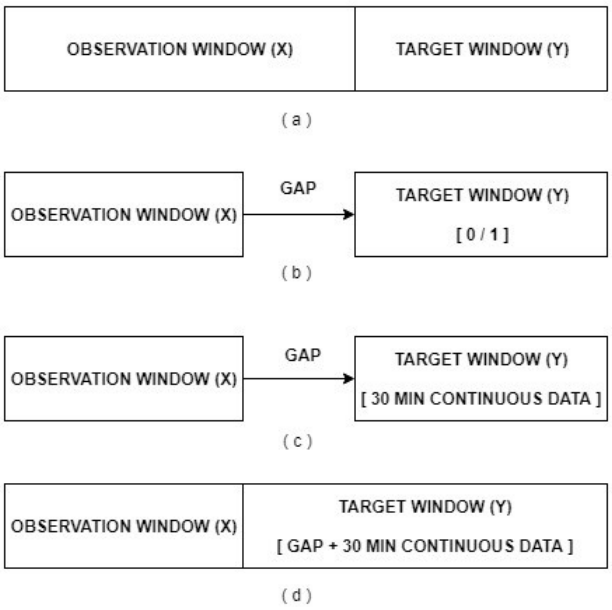


Figure 5.3: Data processing for different algorithms of supervised machine learning

model. The reason behind using only a single observation is that previous studies found that the observation window has minimal impact on model accuracy [Ghosh et al., 2016] [Lee and Mark, 2010b]

- Target window: a target window of 30 minutes was used. This can be called the output of the prediction model. The target window was considered following the definition of the event in e.g. [Irwin and Rippe, 2008].

- Gap window: user-defined gap window which separates the observation and target windows. Gap windows of 0, 5, 10, 15, 30, 60 and 120 minutes are considered here in this work. However, more recent work by Hatib et al. showed that even predicting hypotensive events 15 minutes early could provide clinical significance [Hatib et al., 2018].

Researchers use various definitions of hypotensive and non-hypotensive events. This chapter follows clinical guidelines. A hypotensive event is defined as having MAP values below 60 mmHg for 90% of a 30 minute window [Irwin and Rippe, 2008]. A 30 minute window that does not follow HE is regarded here as a normotensive event.

Data processing differs for the two different supervised machine learning algorithms. For classification, the target window is labelled with a binary classification where 1 represents hypotensive and 0 represents normotensive. An overview of samples used for classification algorithms can be seen in Fig. 5.3 (b). For regression, the target window consists of 30 minutes of continuous MAP. An overview of the samples used for regression can be seen in Fig. 5.3 (c). Fig. 5.3 (d) shows the samples overview that have been used for the regression algorithms. These predict the target window along with the gap window. During supervised data processing, the scripts first look for a definition of HE and then look for a normotensive event. Thus the dataset consists of an equal number of hypotensive and normotensive events. Moreover, all the samples extracted are unique as the predictions of future episodes are omitted when the patient has gone through an ongoing episode. The single compilation mode dataset consists of 21,472 samples with an equal number of hypotensive and normotensive events.

For the bradycardia scenario, the same methodology was applied except to heart rate time series data from the MIMIC III database. The extracted dataset has 16,702 samples, where 4086 samples consist of a bradycardia event and 12616 samples consist of a nonbradycardia event.

5.2.2 Classification Based Prediction of Events

Existing works typically perform classification based prediction unlike the work here. It is therefore useful to gain some understanding of the limitations of the performance of classification based prediction of the events.

Table 5.2: Random forest tuning parameters and ranges

Parameters	Range
n_estimators	[start = 10, stop = 1000]
max_features	['auto', 'sqrt']
max_depth	[10, 110]
min_samples_split	[2, 5, 10]
min_samples_leaf	[1, 2, 4]
bootstrap	[True, False]

Therefore, a number of different classification algorithms of machine learning are considered here. These are used as models for hypotensive event prediction. Classification approaches that are considered here are: Logistic regression, k-nearest neighbour (KNN), decision trees, random forests and multilayer perception (MLP). The reason for this selection of approaches is that these algorithms were found to be effective in different medical applications [Panahiazar et al., 2015, Isler, 2016, Son et al., 2012, Kwon et al., 2018, Narin et al., 2014]. Each model was trained separately for a different combination of observations and gap windows. Initially, default hyperparameter values were used [Pedregosa et al., 2011] for each model. Ten-fold cross-validation was conducted to evaluate classification performance. Evaluation techniques such as classification accuracy, sensitivity, specificity, precision, roc-AUC score were used to evaluate the model performance. The best model was then used for hyperparameter tuning to see whether the accuracy of the model can be improved further. A randomized tuning algorithm [Pedregosa et al., 2011] method was used for hyperparameter tuning of the best performing classifier (random forest). The randomized search is performed over the given parameter set, where each setting is sampled from a distribution over a range of possible parameter values [Bergstra and Bengio, 2012]. Table 5.2 shows the parameters and their ranges that were chosen for hyperparameter tuning. The tuning algorithm considers the grid of hyperparameter ranges and randomly samples from the grid. A ten-fold cross validation was performed with each combination of values. Following hyperparameter tuning, the parameters that were found to give the best accuracy for a 30 minute gap window are:

- n_estimators: 793
- min_samples_split: 2
- min_samples_leaf: 2
- max_features: sqrt
- max_depth: 20
- bootstrap: True

These parameters were then used in the random forest to improve the model accuracy for hypotensive event prediction.

For bradycardia prediction, further hyperparameter tuning was performed. The hypotensive event hyperparameter values were used as the initial set of parameters for bradycardia prediction. Then, following hyperparameter tuning for bradycardia prediction, the parameters found for best accuracy for a 30 minutes gap window are:

- n_estimators: 216
- min_samples_split: 10
- min_samples_leaf: 2
- max_features: sqrt
- max_depth: 50
- bootstrap: False

These parameters were then used in the random forest model to predict bradycardia.

5.3 Results

5.3.1 Hypotensive Event Prediction

5.3.1.1 Hypotensive Event Prediction Classification Results

Results of the classification based prediction models can be seen in Table 5.3. Comparing these different classification algorithms, it appears to show that the random forest prediction models provide the best accuracy compared to logistic regression, KNN, Decission Tree and MLP. The accuracy of 98.12%, 90.24%, 84.95%, 81.92% and 77.59% was found in predicting hypotensive and normotensive events with 0, 5, 10, 15 and 30 minutes in advance respectively. Moreover, the Random Forest model with default parameters is then further tuned with tuning algorithms to explore the potential for an even better outcome. The tuned random forest model was found to improve the forecast accuracy when the gap window increases as can be seen in Table 5.4. For a 30 minute gap window, the default parameter random forest forecast model achieved an accuracy of 77.59%. For the tuned random forest model, an improved accuracy of 78.22% was obtained. Moreover, it is observable that as the gap window increases, the

Table 5.3: Classification results

Model	Evolution Criteria	Gap Window				
		0 Minutes	5 Minutes	10 Minutes	15 Minutes	30 Minutes
Logistic Regression	Accuracy	0.9471	0.8671	0.7998	0.7712	0.7412
	ROC-AUC	0.9757	0.9255	0.8699	0.8435	0.8129
	Sensitivity	0.9586	0.9035	0.8553	0.8433	0.8206
	Specificity	0.9357	0.8306	0.7444	0.6991	0..6616
	Precision	0.9371	0.8421	0.7699	0.7370	0.7081
KNN	Accuracy	0.9367	0.8577	0.8181	0.7977	0.7583
	ROC-AUC	0.9759	0.9208	0.8873	0.8656	0.8274
	Sensitivity	0.9511	0.8423	0.7954	0.7776	0.7422
	Specificity	0.9223	0.8730	0.8407	0.8177	0.7744
	Precision	0.9244	0.8690	0.8331	0.8101	0.7669
Random Forest	Accuracy	0.9812	0.9024	0.8495	0.8192	0.7759
	ROC-AUC	0.9937	0.9580	0.9172	0.8883	0.8439
	Sensitivity	0.9818	0.9180	0.8721	0.8468	0.8083
	Specificity	0.9812	0.8866	0.8273	0.7896	0.7395
	Precision	0.9812	0.8900	0.8347	0.8010	0.7563
Decision Tree	Accuracy	0.9665	0.8498	0.7798	0.7344	0.6814
	ROC-AUC	0.9651	0.8485	0.7779	0.7315	0.6807
	Sensitivity	0.9658	0.8495	0.7785	0.7293	0.6809
	Specificity	0.9660	0.8496	0.7758	0.7339	0.6808
	Precision	0.9660	0.8496	0.7764	0.7327	0.6809
MLP	Accuracy	0.8661	0.7790	0.7315	0.7081	0.7404
	ROC-AUC	0.9562	0.8929	0.8462	0.8208	0.8229
	Sensitivity	0.8712	0.7369	0.7279	0.7248	0.7907
	Specificity	0.8659	0.7986	0.7380	0.6994	0.6795
	Precision	0.8666	0.7853	0.7353	0.7068	0.7116

prediction model loses accuracy. This matches with previous studies of [Lee and Mark, 2010a, Hatib et al., 2018]. It can be understood because predicting further into the future is a more challenging task. For example, in Table 5.3, the random forest has an accuracy of around 98.12% for 0 minute gap window. However, with a larger gap window of 30 minutes, the accuracy decreases to 77.59%. A similar observation can be made for other evaluation metrics, as shown in Table 5.3. Thus, it can be concluded that the gap window has a major impact on model forecast accuracy.

It can therefore be considered that any signal data for the duration of the gap window might play an important role in event prediction. The classification model has to omit this part of the time series to create a gap between the observation window and the target window. Moreover, an algorithm that could analyse these observations in the gap window in predicting the events could aid the forecast model to predict better. The forecasting model with a regression algorithm helps to investigate this scenario further.

Table 5.4: Tuned Random Forest model results

Evolution Criteria	30 minutes gap window
Accuracy	.7822
ROC-AUC	.8523
Sensitivity	.8196
Specificity	.7458
Precision	.7633

5.3.1.2 Hypotensive Event Prediction Regression Results

The regression based forecast model is designed to provide a minute by minute estimate of the MAP values for the period of the target window. Fig. 5.4 shows the results of the forecast model estimating MAP values for the hypotensive event based prediction scenario. The regression model output is compared with the hypotensive event threshold value (60mmHg). The result of this comparison can be directly compared with the classification model output. If the regression model outcome and the true outcome are above the threshold value (60mmHg) then this can be classified as a normotensive event or class 0. If the regression model outcome and the true outcome are below the threshold value (60mmHg) then this can be classified as a hypotensive event or class 1. This can be seen in Figs. 5.4, 5.5, 5.6 and 5.7. These show how the regression model predicted outcome is compared with the true output and the threshold value in order to convert it to a class. For example, the scenarios in Fig. 5.4 are forecasting hypotensive events. The scenarios in Fig. 5.5 are forecasting normotensive events correctly. On the

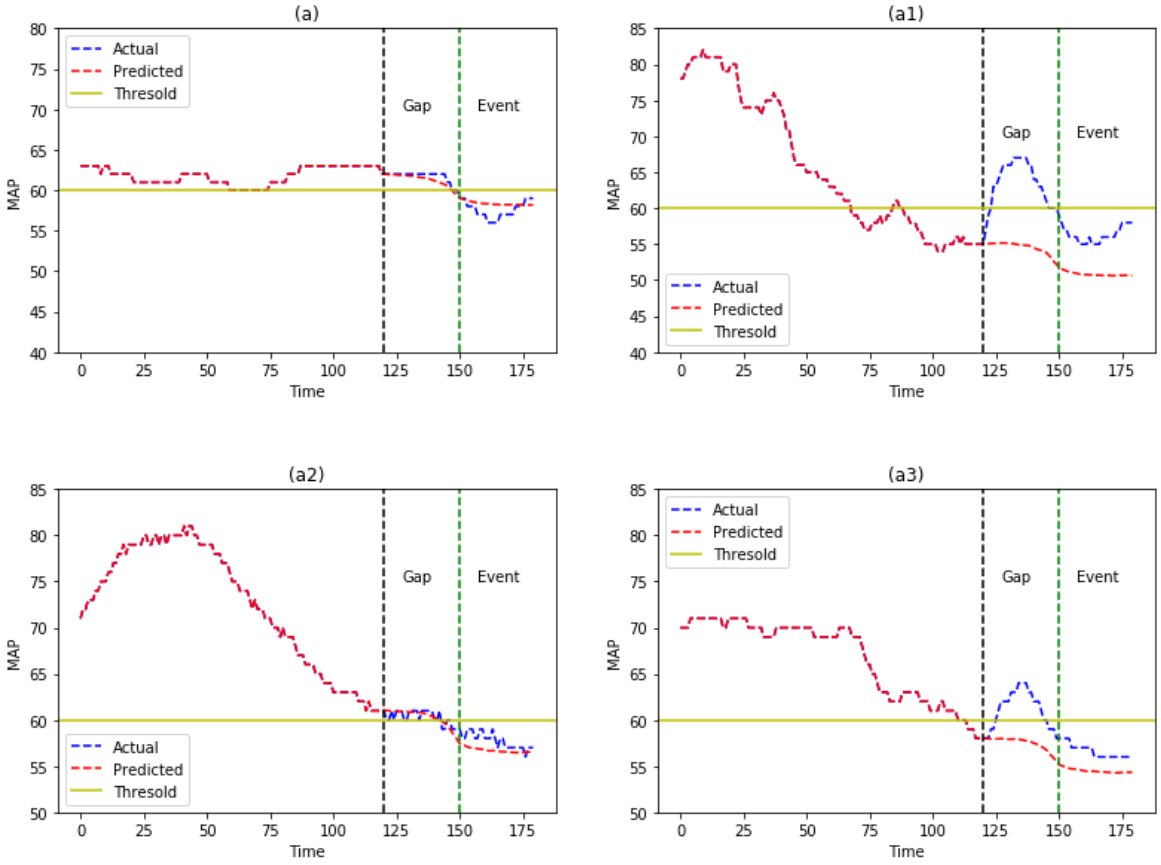


Figure 5.4: Forecast model with regression algorithm (CNN-BISTM) forecasting hypotensive events scenario with 30 minutes gap window shifted towards the target window.

other hand, the scenarios in Fig. 5.7 are incorrectly forecasting normotensive events as hypotensive. In Fig. 5.6, hypotensive events are incorrectly categorised normotensive.

Two different forecast scenarios are considered here. In the first scenario, the forecast model predicts a target window considering the gap as shown in Fig. 5.3(c). Thus, it predicts a target window of 30 minutes. This scenario does not include the observations in the gap window in the forecasting process. Table 5.5 shows the forecast model performance in forecasting events with a user-defined gap window of 0, 5, 10, 15 and 30 minutes. It can be observed from Table 5.5, that the sensitivity, specificity, precision and accuracy decrease gradually as the gap window increases. Model accu-

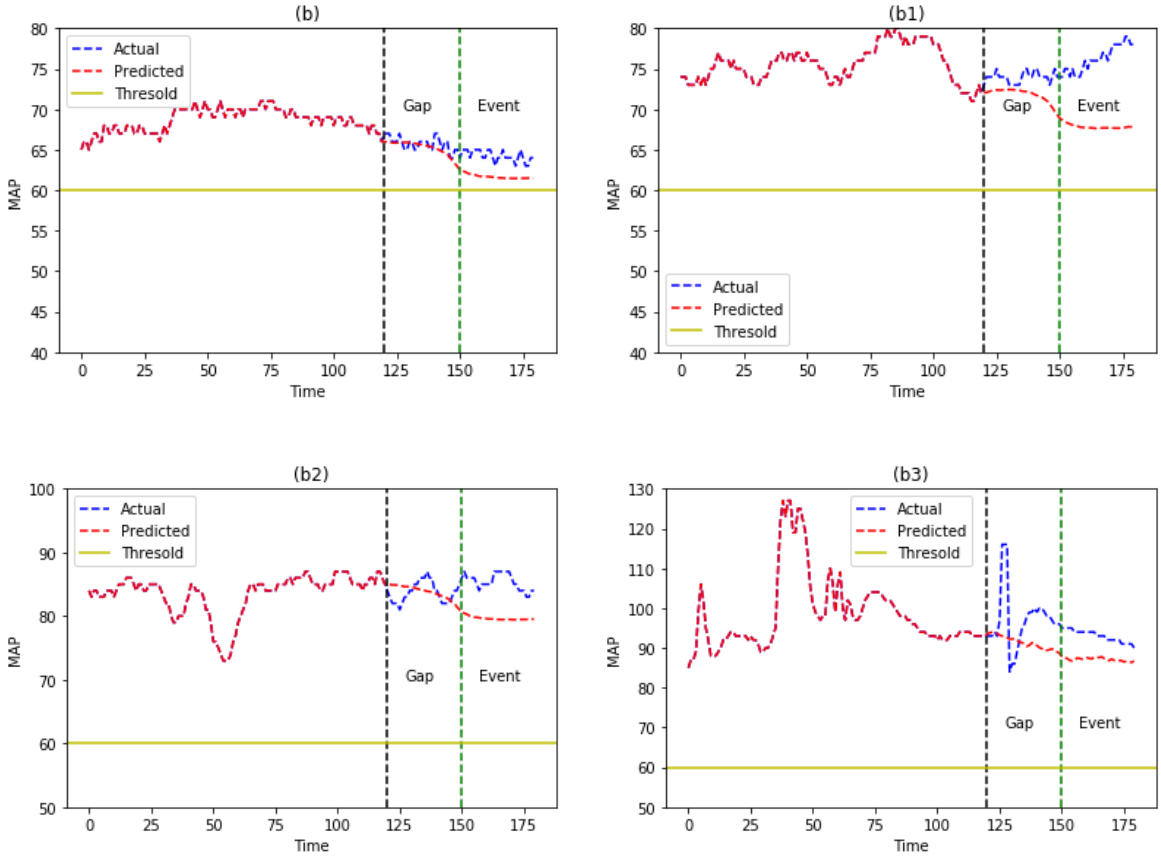


Figure 5.5: Forecast model with regression algorithm (CNN-BISTM) forecasting normotensive events scenario with 30 minutes gap window shifted towards the target window.

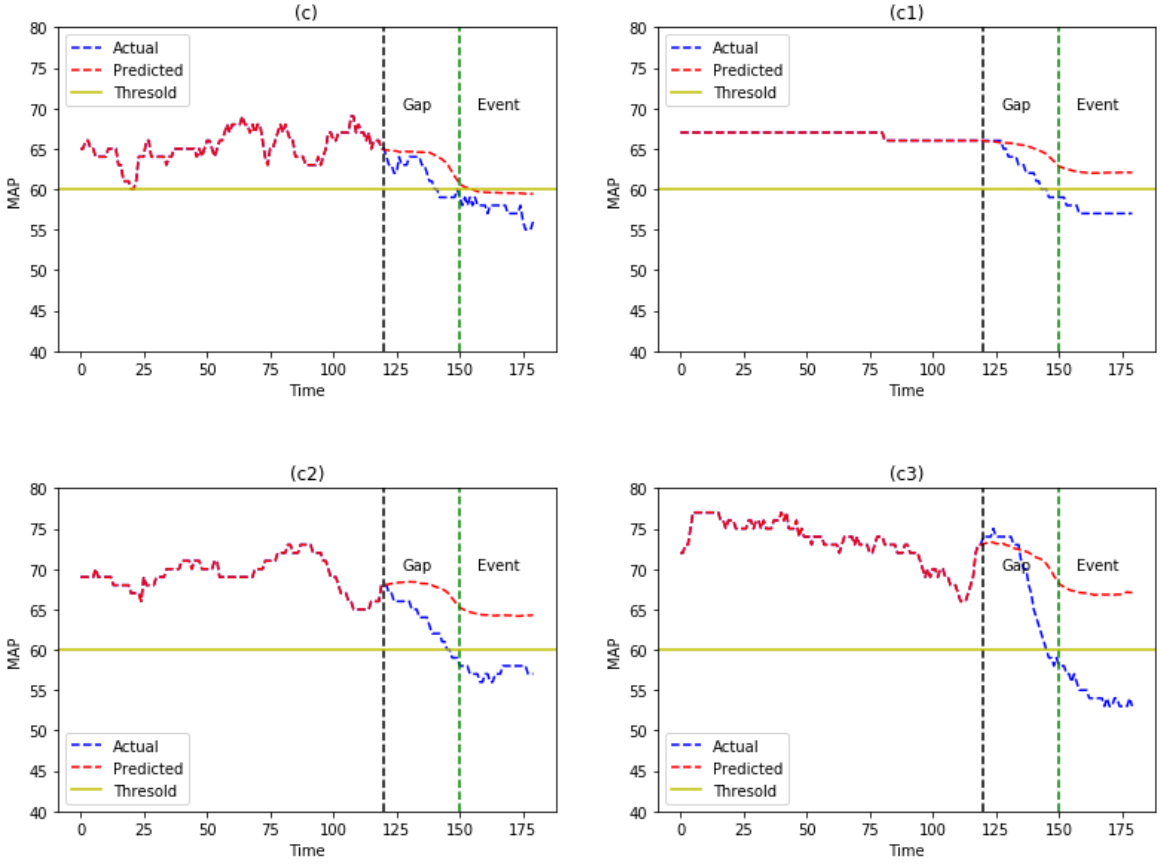


Figure 5.6: Forecast model with regression algorithm (CNN-BISTM) incorrectly forecasting hypotensive events scenario as normotensive events with 30 minutes gap window shifted towards target window.

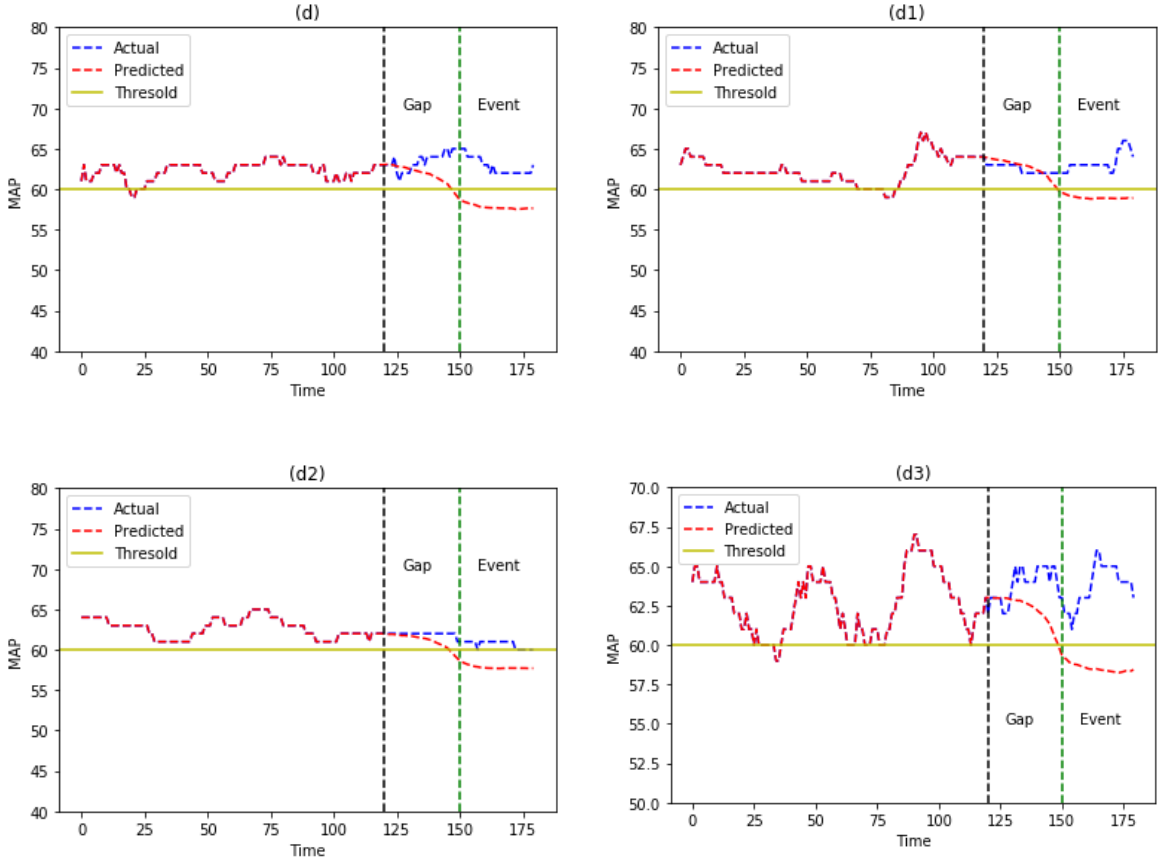


Figure 5.7: Forecast model with regression algorithm (CNN-BISTM) incorrectly forecasting normotensive events scenario as hypotensive events with 30 minutes gap window shifted towards target window.

racy decreases around 39.11% when the gap is increased from 0 to 30 minutes. This matches with previous studies conducted by researchers where they showed that, as the gap window increases, model accuracy decreases [Hatib et al., 2018,Lee and Mark, 2010a]. This also matches with the classification algorithm outcome considered here in this chapter. MAE, MAPE and RMSE increase gradually as the gap window increases. It can also be observed that higher MAE, MAPE and RMSE contribute to the degradation of the model accuracy, sensitivity, specificity and precision, as shown in Table 5.5.

It can also be assumed that the signal data in the gap window is potentially very informative. This is in relation to the intervention of the hypotensive events. Thus it can also be assumed to play a vital role in model performance. However, as noted before, it is not possible to analyse the gap window signal data further when a gap window is considered. In contrast to this, for this second scenario, the gap window is merged with the target window so that the signal is forecast for the gap window and the target window durations, as shown in Fig.5.3(d). Thus the forecast model will predict 30 minutes plus the user-defined gap window.

Results of the second regression scenario can be seen in Table 5.6. These show outstanding improvements in comparison to the first scenario (also in Table 5.5). A comparison between two regression scenarios considering MAE, MAPE and RMSE along with standard deviation are shown in Fig. 5.8. These suggest that the regression algorithm performs better when a 30 minute gap window is added to the target window. Higher MAE, MAPE and RMSE are expected as the forecast horizon is effectively being increased when the gap window is added as shown in Fig.5.3 (d). However, interestingly it can be observed from Tables 5.5 and 5.6 that lower MAE, MAPE and RMSE values are obtained.

For example, when a 30 minute gap window is considered and the model forecast is for the 30 minute target window only; this results in MAE, MAPE and RMSE values of 4.817, 8.047 and 7.595 respectively. This can be compared with the case of when the target window and the gap window are combined so that the prediction model has to forecast for 60 minutes. In such a case, lower MAE, MAPE and RMSE values of 4.088, 6.375 and 5.189 respectively are achieved. The above experiment results appear to suggest that including the gap window signal data with the target window helps

Table 5.5: CNN-BILSTM algorithm results while gap window is considered in forecasting target window.

Metrics	Gap Window in Minutes				
	0	5	10	15	30
MAE	1.494	3.051	3.210	4.335	4.817
MAPE	2.264	5.151	5.362	7.163	8.047
RMSE	2.081	4.842	5.638	6.662	7.595
Sensitivity	.9951	.8575	.7139	.6458	.5717
Specificity	.9908	.8429	.8343	.7717	.6479
Precision	.9907	.8360	.7770	.7234	.7129
Accuracy	.9930	.8500	.7805	.7113	.6019

to create a bridge between past observations and the target window. Consequently, it provides better performance. This is further illustrated with the results presented in Fig. 5.9. This shows when a 30 minute gap window is added to the target window, an accuracy of around 85% is achieved. If the 30 minute gap window is omitted from the forecast, then the regression model provides an accuracy of around 60%. Furthermore, the tuned random forest classifier achieves only 80%. A similar pattern emerges in terms of sensitivity, specificity and precision. Results of different gap window lengths considering accuracy, sensitivity, specificity and precision can be observed in Tables 5.5 and 5.6.

5.3.1.3 Hypotensive Event Prediction with Large Gap Window

A further investigation was conducted with large gap windows of 60 and 120 minutes. The tuned random forest and CNN-BILSTM models were considered. The target window of the CNN-BILSTM model consists of a gap window and events. Table 5.7 and Table 5.8 show the results of the tuned random forest and CNN-BILSTM models. From there it is found that forecasting across a large forecast horizon with the CNN-BILSTM model is found to be difficult. The CNN-BILSTM model provides an accuracy of 68.30% and 66.10% for gap windows of 60 and 120 minutes. This is relatively low compared to the 30 minute gap window accuracy of 85%. This suggests

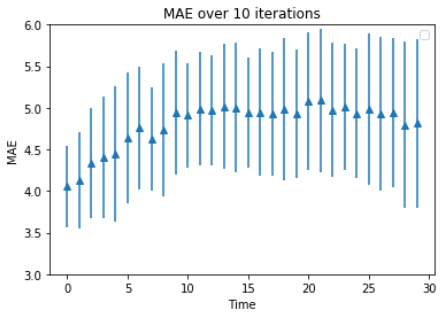
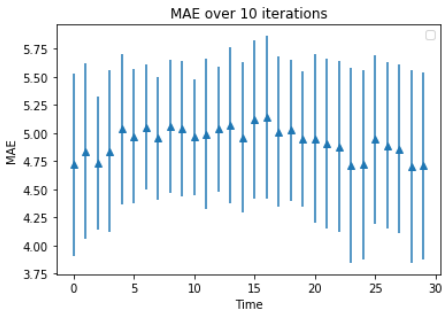
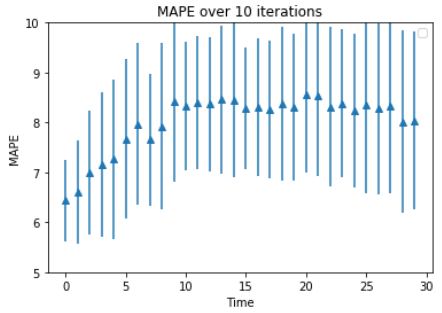
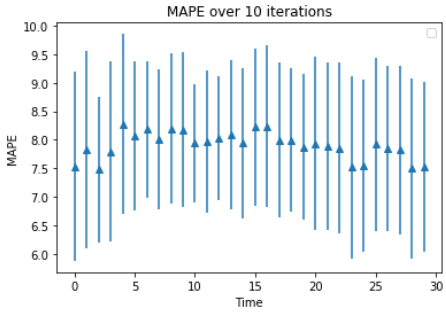
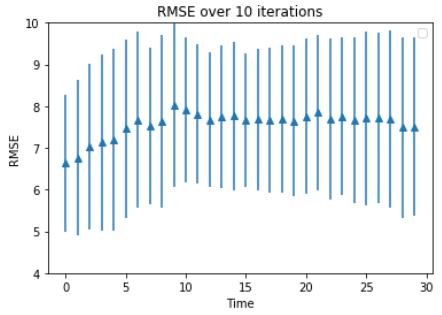
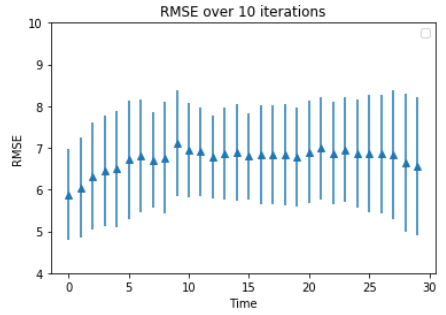
Evalution Metrics	30 Minutes Gap Window	
	Event forecast with gap interval	Event forecast along with gap
MAE		
MAPE		
RMSE		

Figure 5.8: Comparison of two scenario of regression model considering average MAE, MAPE and RMSE along with standard deviation.

Table 5.6: CNN-BILSTM algorithm results while gap window is forecast along with target window

Metrics	Gap Window in Minutes				
	0	5	10	15	30
MAE	1.494	1.652	2.845	3.906	4.088
MAPE	2.264	2.588	4.658	6.214	6.375
RMSE	2.081	2.199	4.293	5.145	5.189
Sensitivity	.9951	.9537	.9352	.9270	.8782
Specificity	.9908	.9578	.9426	.8938	.8250
Precision	.9907	.9543	.9425	.88	.8159
Accuracy	.9930	.9559	.9389	.9090	.85

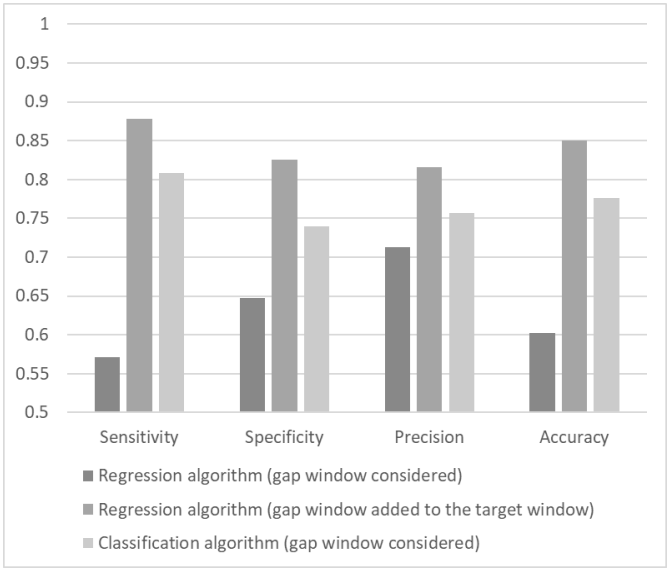


Figure 5.9: Comparison of combined regression algorithm (CNN-BILSTM) with classification algorithm (hyperparameter tuned random forest) considering sensitivity, specificity, precision and accuracy in predicting hypotensive events with 30 minutes gap window

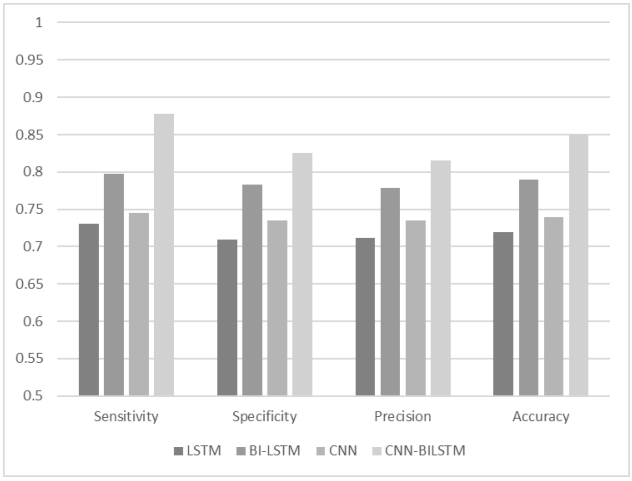


Figure 5.10: Comparison of regression algorithms considering sensitivity, specificity, precision and accuracy in forecasting hypotensive events with 30 minutes gap window

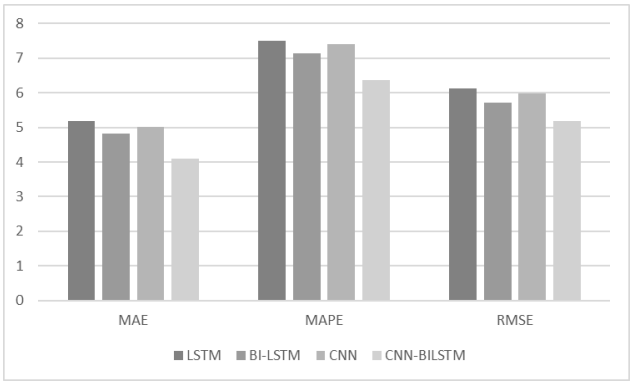


Figure 5.11: Comparison of regression algorithms considering MAE, MAPE and RMSE in forecasting hypotensive events with 30 minutes gap window

the CNN-BILSTM model performs reasonably well for a forecast horizon of up to 1 hour. However forecast horizons beyond 1 hour are not preferable. Moreover, the CNN-BILSTM model is also outclassed by the random forest model when a large gap window is considered. Although it is found that both models perform poorly for these longer gap windows.

Table 5.7: Random Forest results with larger gap window.

Evolution Criteria	Gap window in minutes		
	30	60	120
Accuracy	0.7823	0.7306	0.6946
ROC-AUC	0.8523	0.7990	0.7544
Sensitivity	0.8169	0.7635	0.7198
Specificity	0.7458	0.6977	0.6713
Precision	0.7633	0.7164	0.6865

Table 5.8: CNN-BILSTM results with larger gap window.

Evolution Criteria	Gap window in minutes		
	30	60	120
MAE	4.088	6.001	6.955
MAPE	6.375	9.324	10.773
RMSE	5.189	9.203	10.159
Sensitivity	.8782	.7160	.6747
Specificity	.8250	.6473	.6485
Precision	.8159	.6869	.6446
Accuracy	.85	.6830	.6610

5.3.2 Bradycardia Prediction Scenario

Heart rate time series analysis has been conducted following the same methodology used for the hypotensive event prediction. Heart rate time series analysis was performed to

predict Bradycardia. Heart rate time series data was extracted following the same data extraction and preparation used in the methodology section.

5.3.2.1 Classification Results

Results of the tuned random forest prediction model in predicting bradycardia can be seen in Table 5.9. Results appear to show that it is possible to predict bradycardia with reasonable accuracy. However, model performance decreases as the gap window increases. For a 5 minute gap window, the model achieves an accuracy of 93.86%. For a 120 minute gap window, the model performance decreases to 81.05%. An appropriate balance between sensitivity and specificity can be seen for a short gap window, but it starts to vanish with a large gap window.

Table 5.9: Random Forest results in predicting Bradycardia.

Evolution Criteria	Gap window in minutes				
	5	15	30	60	120
Accuracy	.9386	.8961	.8545	.8348	.8105
ROC-AUC	.9778	.9445	.9349	.9225	.9055
Sensitivity	.9121	.8367	.7971	.7687	.7420
Specificity	.9482	.9153	.9140	.9080	.8996
Precision	.8509	.7619	.7501	.7302	.7054

5.3.2.2 Regression Results

Results of the tuned CNN-BILSTM model in predicting bradycardia can be seen in Table 5.10. These results show that the CNN-BILSTM model can forecast well with low RMSE, MAE and MAPE values observed. For a 5 minute gap window, the model forecasts the forecast horizon with MAE, MAPE and RMSE values of 2.884, 4.047, 4.678 respectively. However, as expected, the MAE, MAPE and RMSE increase for longer forecast horizons. The model achieves MAE, MAPE and RMSE values of 5.615, 7.649, 8.519 respectively for a 120 minutes gap window. Lower values of MAE, MAPE and RMSE help to improve event prediction accuracy. The model achieves an accuracy of 94.30% for a 5 minute gap window. This is when MAE, MAPE and RMSE values are

relatively low. For a greater gap window, the model achieves an accuracy of 75.80% for the higher MAE, MAPE and RMSE values. This also suggests that a longer forecast horizon increases error in forecasting and leads to decreasing accuracy. Some exemplar scenarios are demonstrated in Fig. 5.12 where forecasting of bradycardia is identified correctly. Fig. 5.13 shows forecasting of non-bradycardia events correctly. However, the scenarios in Fig. 5.15 incorrectly forecast non-bradycardia events as bradycardia and Fig. 5.14 incorrectly forecast bradycardia events as non-bradycardia.

Table 5.10: CNN-BILSTM results for the prediction of Bradycardia.

Evolution	Gap window in minutes				
Criteria	5	15	30	60	120
MAE	2.884	3.615	4.032	4.206	5.615
MAPE	4.047	5.268	5.615	6.035	7.649
RMSE	4.678	6.310	6.482	6.910	8.519
Sensitivity	.8358	.8139	.7167	.6758	.5939
Specificity	.9863	.9672	.9360	.8486	.8362
Precision	.9612	.9217	.8147	.6681	.6336
Accuracy	.9430	.9180	.8743	.7950	.7580

5.4 Discussion

The results overall have helped to illustrate that the proposed regression algorithm could potentially be trained and used to intervene in cases where hypotensive and bradycardia events might occur. The addition of the gap window information during the training process can be seen to help to improve the forecast model performance. However, this is only true for a gap window of up to 30 minutes. The forecast model is unable to provide better accuracy when a gap window goes above 30 minutes. This is because with more than 30 minutes, the gap window model has to forecast for a long time. This suggests forecasting of a long horizon is a difficult task. The forecast model with 0 minute gap window is able to forecast events with more than 99 per cent accuracy. Such a scenario in itself could be informative to a health care professional

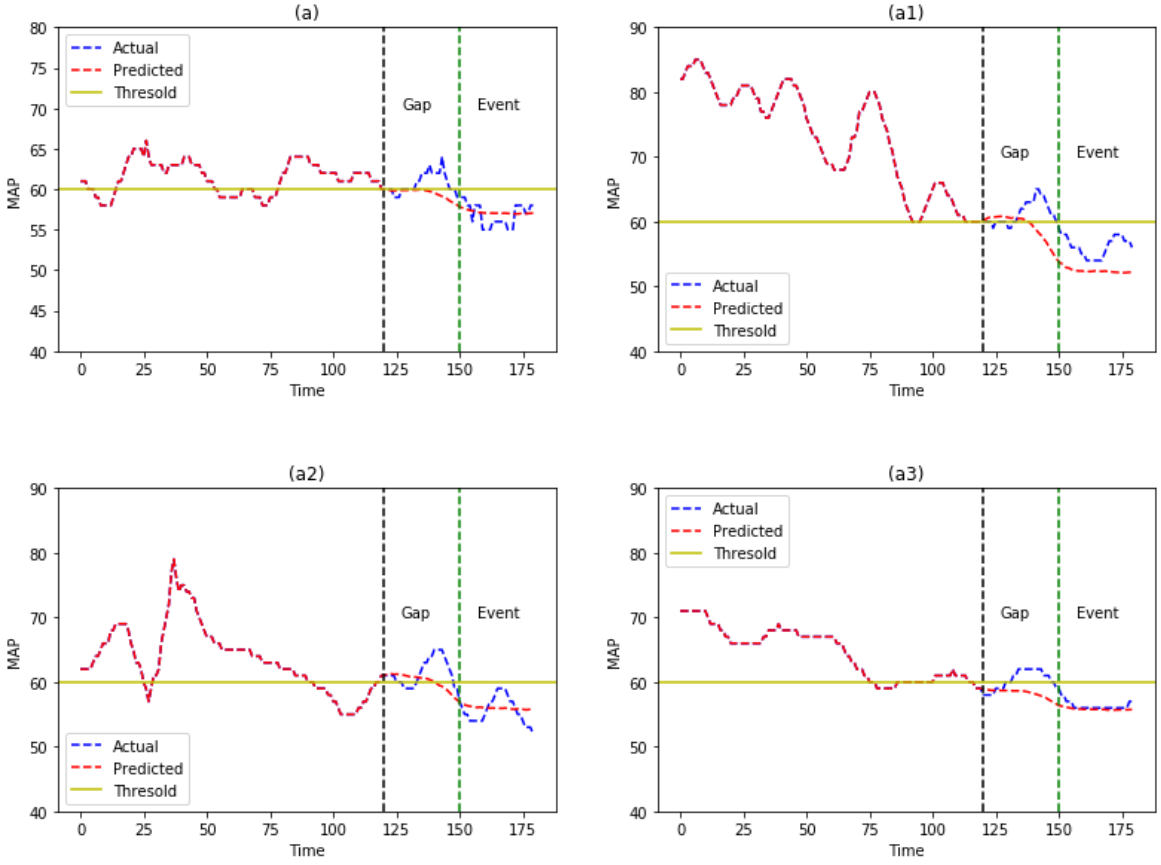


Figure 5.12: Forecast model with regression algorithm (CNN-BISTM) forecasting bradycardia events exemplar scenarios with a 30 minute gap window combined with the target window.

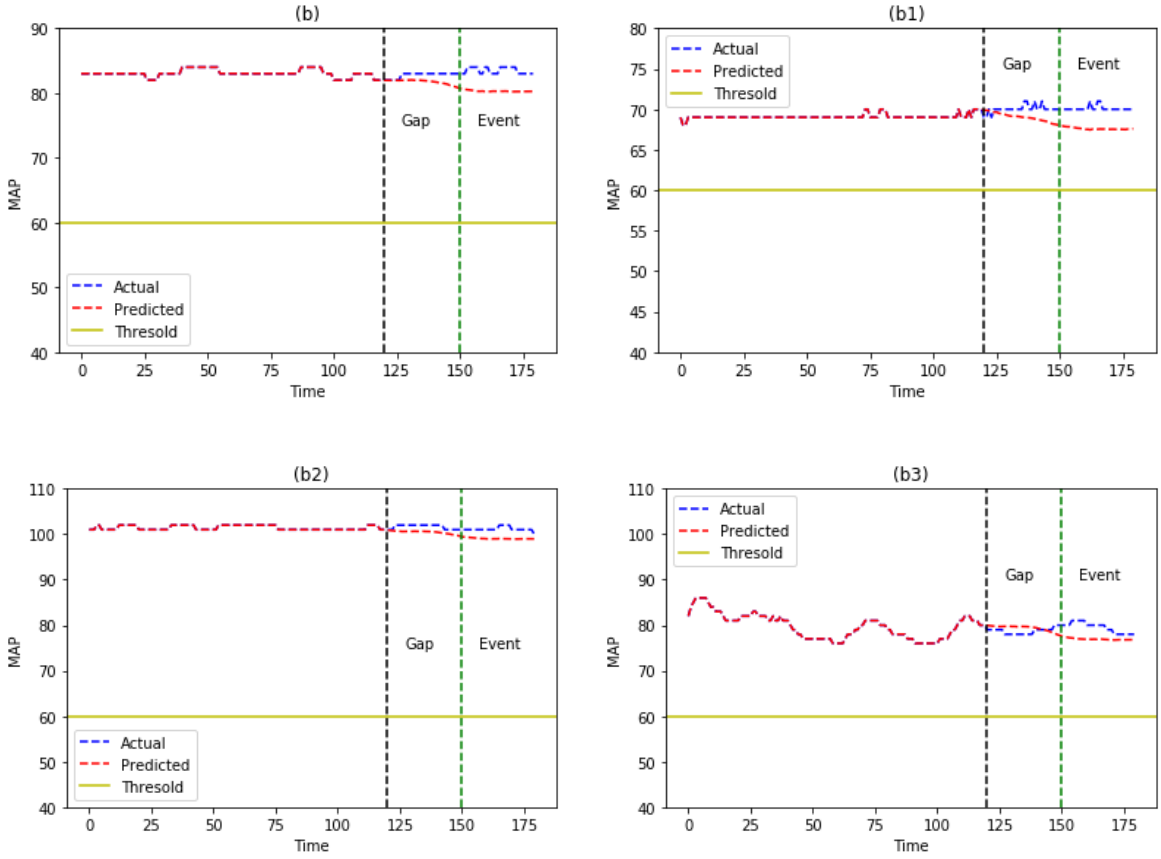


Figure 5.13: Forecast model with regression algorithm (CNN-BISTM) forecasting non-bradycardia events exemplar scenarios with a 30 minute gap window forecast with the target window.

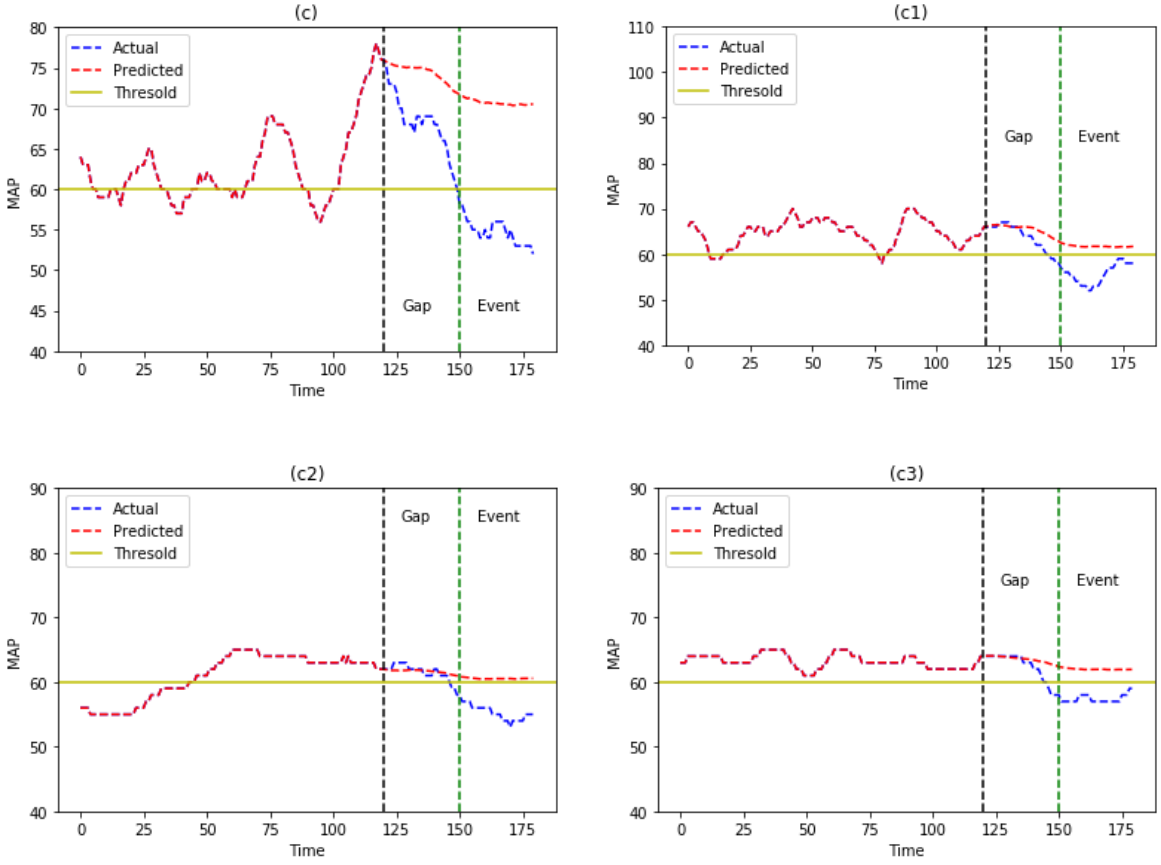


Figure 5.14: Forecast model with regression algorithm (CNN-BISTM) incorrectly forecasting bradycardia with a 30 minute gap window (forecasting in both the gap and target windows).

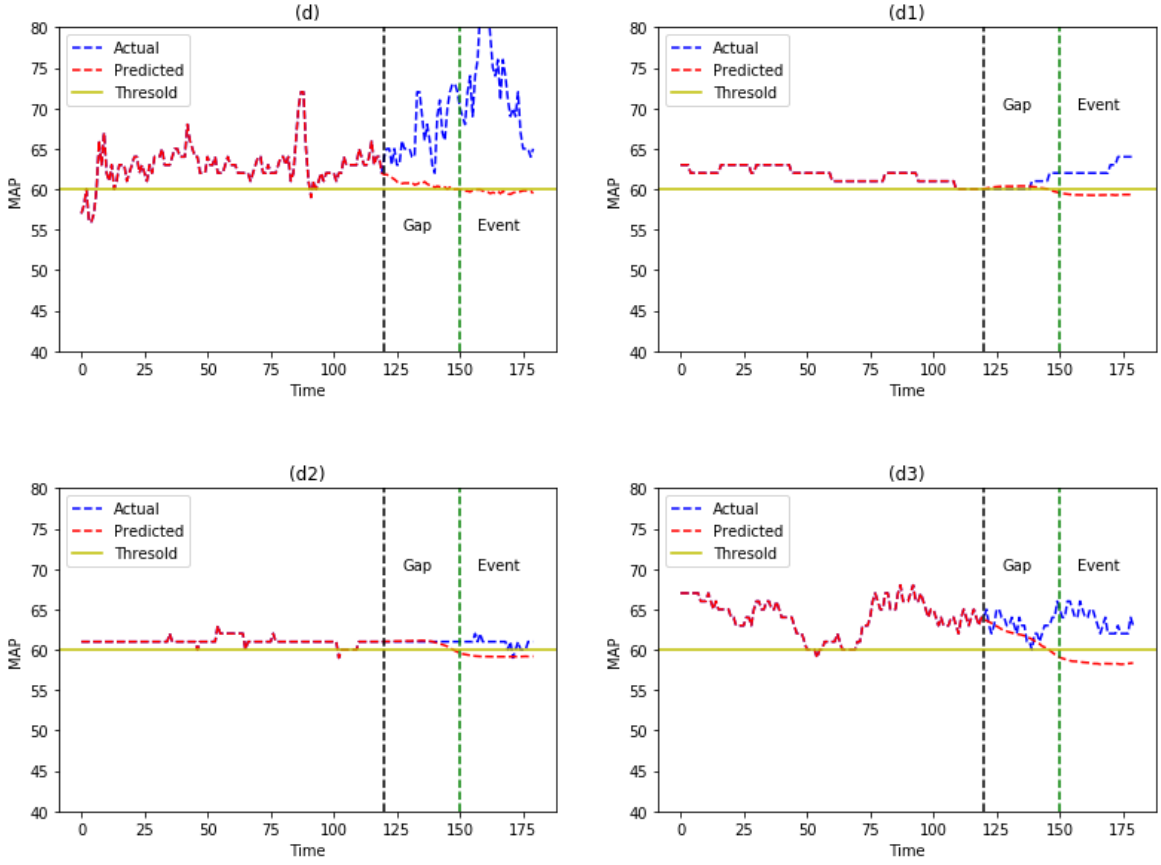


Figure 5.15: Forecast model with regression algorithm(CNN-BISTM) incorrectly forecasting non-bradycardia events as bradycardia events exemplar scenarios with 30 minute gap window and target window.

as it is found that risks and complications due to the aforementioned events begin to develop shortly after events [Ghosh et al., 2016]. Model forecasting with a 30 minute target window or as an event means that the model is able to predict when it is going to start. This allows health care professionals almost 30 minutes to react even there is no gap considered.

The comparative analysis appears to show that the regression algorithm does have the edge over the classification algorithm for event prediction. This is true for a gap window of up to 30 minute. However, regression performs worse than the classification if the gap window is not included in the forecast. Fig. 5.9 includes a comparison of the prediction models when hypotensive events are intervened 30 minutes in advance. Sensitivity, specificity, precision and accuracy are used to help to evaluate them. It has been observed that the classification model performs better than the regression algorithm when a gap window is omitted in event prediction. When a 30 minute gap window is omitted, the regression model achieves an accuracy of around 60%. This is very low compared to the classification model accuracy of 77.59%. However, it can be seen that the forecast model with regression algorithm achieves the highest accuracy of 85% when a 30 minutes gap window is added to the target window. The classification algorithm achieves an accuracy of 77.59% in intervening events 30 minutes in advance.

The regression model sensitivity, specificity and precision values are given as .8782 .8250 and .8159 respectively in predicting hypotensive events with 30 minutes gap window. This is in comparison to the classification model values of .8083, .7395 and .7563 respectively. Moreover, the prediction of bradycardia results also support the above claim. From Table 5.9 and 5.10 it can be seen that the regression model performs better than the classification model. Again this is true until a gap window of longer than 30 minutes is considered. Moreover, the output of the regression algorithm model does provide the continuous outcome of the events rather than categorical values. Healthcare professionals might be able to extract more information from the continuous outcome of the events in the decision making process.

Comparison between the different regression algorithm presented in Fig. 5.10 suggest that combined CNN-BILSTM regression algorithm performs better than LSTM, BILSTM and CNN algorithms individually. The forecast model with LSTM, BILSTM and CNN algorithms were tuned following the same methodology applied to tuning

the CNN-BILSTM model. The CNN-BILSTM forecast model achieved the highest accuracy of 85% in comparison to LSTM 72%, BILSTM 79% and CNN 74% accuracy. Forecast models with the CNN-BILSTM algorithm also achieve higher sensitivity, specificity and precision compared to the other mentioned algorithms. Moreover, MAE, MAPE and RMSE were also taken into consideration to compare different regression algorithms represented in Fig. 5.11 . It is observable that the forecast models with the CNN-BILSTM algorithms provide lower MAE, MAPE and RMSE compared to the LSTM, BILSTM and CNN algorithms.

Hatib et al. showed that machine learning algorithms could be trained to predict hypotensive events [Hatib et al., 2018]. Their work predicts hypotensive events, respectively, with a sensitivity and specificity of 88% and 87% 15 minutes before; 89% and 90% 10 minutes before; 92% and 92% 5 minutes before. In comparison, the CNN-BILSTM model represented here in this chapter predicts events with a sensitivity and specificity of 92.70% and 89.38% 15 minutes before; 93.52% and 94.26% 10 minutes before; and 95.37% and 95.78% 5 minutes before. This chapter also presented a sensitivity and specificity of 87.82% and 82.50% 30 minutes before the events.

Ghosh et al. predicted events using sequential contrast patterns [Ghosh et al., 2016]. They showed that the algorithm predicts events an hour before with the highest accuracy of 83.54%, where sensitivity and specificity are recorded as 100% and 68.29%. Their work suggests that the algorithm can predict the hypotensive event with 100% accuracy for the datasets extractable from the link (`https://github.com/s-ghosh/hypotension`). These datasets were extracted and fed into the tuned Random Forest model and the CNN-BILSTM model described in this chapter. It was found that both the random forest and CNN-BILSTM model predict the events with 100% accuracy. The extracted datasets are then investigated further and found deprecated.

Lee and Mark in their work with classification obtained an accuracy of around 76% with sensitivity and specificity of 76% and 75% with a 60 minute gap window [Lee and Mark, 2010b]. Moreover, their work also included a regression approach which showed the best mean absolute error of 9.67. In contrast to this, the CNN-BILSTM model described here obtained a mean absolute error of 4.088 with 30 minute gap window. Furthermore they did not consider how the forecast signal could be used to detect events. The top performers from 2009 Computers in Cardiology competition [Moody

and Lehman, 2009] was Chen et al. [Chen et al., 2009]. They obtained, with a 30 minute timeline, an accuracy of 82.5% correct predictions. However the test dataset in that challenge was limited to only 40 samples. The CNN-BILSTM forecast model presented here provides 85% accuracy but on a much larger test sample for a 30 minute timeline.

5.5 Limitations

The datasets used here in the described experiments have 21,472 samples for hypotensive event prediction and 16702 samples for bradycardia prediction. The machine learning model could be trained better with a dataset consisting of more samples. This could then help to improve model accuracy. Moreover, univariate time series data has been considered in these results. It would be interesting to see how multivariate time series data could perform under similar scenarios. Moreover, most of the works conducted by other researchers have not shared any univariate or multivariate datasets [Lee and Mark, 2010b] [Hatib et al., 2018]. This makes it harder to compare the results. A standard dataset with a large number of samples in it would be preferable. The Computers in Cardiology competition [Moody and Lehman, 2009] provides a platform for this but with limited samples which are less suitable for machine learning.

5.6 Summary and Concluding Remarks

This chapter has investigated the scope of supervised machine learning in hypotensive and bradycardia event intervention. This was performed by analysing MAP and HR time series data. The investigation was conducted on both classification and regression algorithms of supervised machine learning. Results from the study appear to show that supervised machine learning does have the capability to intervene in events in advance with reasonable accuracy. It is also observed that the forecast model with a regression algorithm does perform better when the gap window of up to 30 minutes is added to the target forecast window. This was compared with the regression that performed poorly when the gap interval was omitted from the gap information. It can be tentatively concluded, from the experiments that signal data in the gap window plays

an important role. However, existing prediction models with a classification algorithm can not utilise the information in the gap window. Instead prediction models with that make use of regression can directly use the gap window information. Most importantly, it is found that inclusion and analysing these observations in the gap window does help to improve the model accuracy.

Chapter 6

Conclusions and Future Work

This chapter presents the conclusions to the research described in this thesis. It also includes summaries of the investigations and experiments that were performed in order to explore the research problems and objectives of this thesis. This chapter also includes consideration of the broader research agenda of physiological time series analysis. As part of this, some potential future research directions are provided.

6.1 Conclusions

Patients' physiological time-series data (e.g. blood pressure and heart rate) are recorded through advanced and specialised medical equipment in Intensive Care Units (ICU) in hospitals. More importantly, researchers have been given access to captured examples of these data through rich databases such as the MIMIC databases [Johnson et al., 2016a]. Researchers can use machine learning techniques to handle complex medical data and to help search and make sense of the data [Clifton et al., 2015,Schaefer et al., 2005]. In tandem to this, machine learning algorithms and methodologies have received quite a bit of attention in the prediction and classification of physiological data [Lipton et al., 2015b] [Nguyen et al., 2017] [Zhu et al., 2018]. However little attention has been given to the forecasting of physiological data with machine learning techniques in medical applications. The primary goal of the research described here was to explore the scope of machine learning algorithms in analysing blood pressure and heart rate time series data for forecasting. This lead to clinical event prediction such as hypotensive

and bradycardia events. The experiments of the research are based on time-series data from the MIMIC databases.

First, a relatively detailed yet broad background chapter covered a range of different topics. This considered different aspects including machine learning, model evaluation procedures and metrics of time series prediction. Then, a literature review was included in chapter 3. The literature review focused on the application of physiological time series data with medical applications. This was useful as it helped to identify the current methodologies used in the prediction of physiological time series data in the medical domain. This helped to add insight into how existing methods predict physiological time series data and also in terms of their shortcomings. The results of this literature review suggested that applications of physiological time series data have been extensively used for classification in medical problems but rarely used for actual forecasting purposes. Although it is found that time series forecasting has been useful for numerous other applications such as solar energy, economics and electric load forecasting. This provided encouragement to investigate time series forecasting of physiological data and the medical applications such as hypotensive and bradycardia event predictions.

Time-series physiological data (BP and HR) were extracted from MIMIC II and III databases. These time series have been processed according to the requirements of the forecasting and prediction models. The sliding window technique was used to process the raw data and to convert it into a form appropriate for supervised learning. This was the requirement of the prediction model. The overall process of data preparation is described in chapter 2.

An investigation was performed using a number of forecasting strategies. These are required for forecasting and are typically combined with the machine learning algorithms described in chapter 4. Forecast models were built by combining machine learning algorithms and forecast strategies. The aim was to explore the best possible forecast strategies, algorithms and their combinations. Results from the investigation appeared to suggest that MIMO and DIRMO strategies are the best forecast strategies. This was compared to traditional strategies in forecasting applied to forecasting of BP and HR 30 minutes in advance. Moreover, it was also shown that the BI-LSTM algorithm forecasts physiological data better than the traditional LSTM algorithm.

The reason to choose LSTM and BI-LSTM algorithms is that they have been designed and found to be useful in long term sequential forecasting. This helps to overcome the limitations of existing work mentioned in Section 1.2 and chapter 3 in terms of a lack of comparisons between differing forecast strategies and machine learning algorithms. Moreover, the work described here included investigations in terms of forecasting blood pressure time series for both single-variate and multivariate cases. This is where the heart rate signal can be simultaneously considered along with the inter-related blood pressure to help forecast future values of blood pressure. This is referred to here as the multivariate case. Forecast models were built by combining machine learning algorithms together with the two best forecast strategies identified in chapter 4. This helped to overcome some limitations with existing work as mentioned in Section 1.2 and chapter 3. There, it was shown that the current literature is missing comparisons of data modelling approaches (i.e. in terms of the univariate and multivariate cases) and also in terms of the machine learning algorithms. The quantitative results obtained here appear to show that the best forecasting model is the BI-LSTM model making use of multi-step forecasting and multivariate data. Further analysis on the bounds of the RMS error on this winning model and strategy is also undertaken. This appeared to show a somewhat straight line relation between forecast horizon time and the RMS error giving a slope of 0.1. The above investigation addresses the first research problem mentioned in chapter 1 and the first three objectives of the research mentioned in Section 1.3.

Finally, time series analysis of blood pressure and heart rate have been conducted using both regression and classification algorithms. They were compared here in the context of investigating two medical applications. These were for the early intervention of hypotensive events and bradycardia events. The main goal of these experiments was to examine whether the regression algorithm can help in the intervention of future hypotensive events or future bradycardia. This helps to address the second research problem mentioned in chapter 1 and the final objective of the research mentioned in Section 1.3.

Different classification algorithms were also compared against each other. The random forest was found to have the best performance in terms of accuracy, ROC-AUC, sensitivity, specificity and precision. Different regression algorithms were also

examined and the CNN-BILSTM model was found to provide the best performance in terms of RMSE, MAE, MAPE, accuracy, sensitivity, specificity and precision. This helped to address limitations in terms of the existing literature discussed. It was shown in Section 1.2 and chapter 3 that existing literature is missing some potentially important combinations of machine learning algorithms when forecasting physiological time series data. It was found from the subsequent classification results obtained from the research presented here, that the gap window plays a vital role in model performance. Results showed that as the gap window increases, accuracy decreases. This indicates that the observations in the gap window contain critical information. Thus, inclusion of these observations in time series analysis could potentially help to improve model performance. Moreover, models were tuned with tuning algorithms to find the best tuning parameters. This also helped to address another limitation of existing work as mentioned in Section 1.2 and chapter 3 where it was found that hyper-parameter tuning was not considered during model building procedure.

Following this, the gap window and target window were combined together to form a single forecast window. This meant that the model will forecast both the gap window information and the target window. Results from the CNN-BILSTM model indicate that the model can provide excellent performance. However, it is found to be limited up to a 30 minute gap window (combined with the target window). This suggests the model can forecast across a 1 hour forecast horizon (30 minute gap window and 30 minute target window) reasonably well. It was found that the model accuracy reduces as the gap window length is increased to more than the 30 minutes.

The forecast model with a 0 minute gap window forecasts events with around 99 per cent accuracy. This scenario alone can be encouraging for the potential application in a health care setting. Model forecasting with a 30 minute target window means that the model is able to predict events and in particular when they are going to start. This would give a health care professional a notice of around 30 minutes to react or to implement preventive treatments. This is even when there is no gap window. Unfortunately, risks or complications are often possible when the events begin to develop or shortly after. Thus it can be tentatively concluded that the forecasting model with the regression algorithm does have the scope to predict dangerous clinical events and assisting healthcare professional.

The significant findings from the experiments which are believed to be informative to fellow researchers and forecast model developers are listed below:

- Time series forecasting of physiological data (i.e. blood pressure and heart rate) with temporal dependency can be used to intervene in dangerous clinical events. Here in this research, the time series of blood pressure and heart rate have been forecast using a hybrid CNN and BI-LSTM model for regression. Forecast values were used to successfully detect hypotensive and bradycardia events with reasonable accuracy. This research also shows that the performance of the regression based forecast model has better performance in comparison to a classification based prediction model. This is true for the detection of events with up to a 30 minute gap window.

- The forecast model predicts the forecast horizon with a very low RMSE when there is no gap considered between the observation and target windows. This could potentially be of interest to healthcare professionals. This is because forecasting events with a 0 minute gap window means that the model is predicting the event before it starts to happen. Here in this research, the forecast model is able to identify hypotensive and normotensive events with an accuracy of around 99 per cent. This is when there is no gap between the observation window and the target window. This can be seen in Table 5.6.

- The prediction model with a regression algorithm also provides a continuous stream of values over the forecast horizon. It could be assumed that this carries extra information. A purely classification based prediction model only provides a single binary result. However, for a forecast model that forecast over the entire target window could also potentially aid healthcare professionals in decision making.

- The gap window plays a vital role in critical event prediction. The observation in the gap window has been found to be useful in critical event forecasting. Forecasting the gap window observations along with event observations helps to improve model performance.

- The investigation of machine learning algorithms appears to show that the BI-LSTM algorithms outperform the traditional LSTM and CNN. However, it was also found that the combination of CNN and BILSTM is more effective in forecasting physiological time series data.

- Investigation of forecast strategies suggests that DIRMO and MIMO are the two best forecast strategies. They could be used in combination with machine learning algorithms in forecasting physiological time series data.

6.2 Limitations

Some limitations of the research are described below:

- In chapter 4, blood pressure and heart rate time series data have been forecast. Forecast models were built combining different machine learning algorithms, forecast strategies and data approaches. However, no hyperparameter tuning was considered for these models. This was to investigate the feasibility of different forecast strategies and different data techniques, rather than medical applications. Thus, a standard parameter set was used for all models. However, hyperparameter tuning was considered for forecasting models in chapter 5, where blood pressure and heart rate were forecast to help in predicting critical events.

- In chapter 4, Experimental datasets were selected from the hypotension group of an ICD-9 code. This is because we wanted to investigate machine learning techniques in forecasting low blood pressure, which leads to a hypotensive event. However, other subgroups such as patients' age, sex, ethnic groups, medications, disease were not considered while modelling. Considering specific or combinations of subgroups could help to improve model accuracy.

- In chapter 5, univariate time series data were considered in forecasting hypotensive and bradycardia events. However, in chapter 4, it was found that a model that included multivariate data performed better than the univariate data in forecasting blood pressure. The multivariate data were not considered in chapter 5 because of difficulties in extracting the multivariate data. In multivariate data,

data points of all variables need to be measured in the same period. These are less common in available data, and such long datasets are infrequent and harder to extract. Moreover, extracting features out of multivariate data is challenging.

6.3 Future Work

This thesis is the first step towards a broader research agenda that aims at developing forecast models. Physiological time-series data were analysed with machine learning algorithms, forecast strategies and different data approaches. Potential directions for future research of this current work are listed below.

- From this study and existing literature, it is quite clear that observations in the gap window play an important role in model performance in predicting critical events. A standard definition of the gap window pattern before the events would help in the analysis of the data points in the gap window. Thus, it would be expected to eventually help to prevent the events. This would mean that instead of predicting the events, predicting of the underlying patterns available in the gap window would be a more insightful and accurate method. This in turn would lead to the more accurate event detection.

- One of the challenges was to look for and prepare a dataset for the developed models. For machine learning, datasets are readily available for the experiments when considering general forecasting application areas like electric load, stock exchange, currency rate. However, it is rare to find a dataset that consists of physiological time series data for forecasting purposes in medical applications. The few papers that have experimented with such datasets have not shared their data. Considering this, the development of a dataset and to make it available for research purposes would benefit the research community as a whole.

- Multivariate time series data could be used in forecasting hypotensive and bradycardia events. It was found that multivariate data does help to improve the forecast accuracy by exploring the correlation between multiple variables, as shown in chapter 4. However, this would require a rich dataset consisting of numerous variables with temporal dependencies. Building such a dataset and exploring a

methodology to extract important feature would also provide a significant contribution.

- It would be interesting to see how the models behave by considering contextual information. Information such as Body Mass Index (BMI), age, gender along with physiological time series data in forecasting critical events could provide a richer, better informed set of variables.

- The main aim of the developed forecast models was to forecast physiological time series data. However, the described methodology can be used for a wide range of other medical applications. The forecast models may also be applicable to more general forecasting applications where time series forecasting is applied.

Bibliography

[Abdeljaber et al., 2017] Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., and Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration, 388:154–170.

[Aboagye-Sarfo et al., 2015] Aboagye-Sarfo, P., Mai, Q., Sanfilippo, F. M., Preen, D. B., Stewart, L. M., and Fatovich, D. M. (2015). A comparison of multivariate and univariate time series approaches to modelling and forecasting emergency department demand in western australia. Journal of biomedical informatics, 57:62–73.

[Aczon et al., 2017] Aczon, M., Ledbetter, D., Ho, L., Gunny, A., Flynn, A., Williams, J., and Wetzel, R. (2017). Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks, 2017. arXiv preprint arXiv:1701.06675.

[Aggarwal, 2014] Aggarwal, C. C. (2014). Data classification: algorithms and applications. CRC press.

[Almeida and Nabney, 2016] Almeida, V. G. and Nabney, I. T. (2016). Early warnings of heart rate deterioration. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 940–943.

[Angner et al., 2009] Angner, E., Ray, M. N., Saag, K. G., and Allison, J. J. (2009). Health and happiness among older adults: a community-based study. Journal of health psychology, 14(4):503–512.

[Austin et al., 2013] Austin, P. C., Tu, J. V., Ho, J. E., Levy, D., and Lee, D. S. (2013). Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes. Journal of clinical epidemiology, 66(4):398–407.

[Baccouche et al., 2011] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., and Baskurt, A. (2011). Sequential deep learning for human action recognition. In International workshop on human behavior understanding, pages 29–39. Springer.

[Bagley et al., 2001] Bagley, S. C., White, H., and Golomb, B. A. (2001). Logistic regression in the medical literature:: Standards for use and reporting, with particular attention to one medical domain. Journal of clinical epidemiology, 54(10):979–985.

[Bassale, 2001] Bassale, J. (2001). Hypotension prediction arterial blood pressure variability. Tecnhical report.

[Bellazzi et al., 2011] Bellazzi, R., Ferrazzi, F., and Sacchi, L. (2011). Predictive data mining in clinical medicine: a focus on selected methods and applications. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(5):416–430.

[Bellazzi et al., 2005] Bellazzi, R., Larizza, C., Magni, P., and Bellazzi, R. (2005). Temporal data mining for the quality assessment of hemodialysis services. Artificial intelligence in medicine, 34(1):25–39.

[Bengio et al., 1994] Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks, 5(2):157–166.

[Bennett and Hauser, 2013] Bennett, C. C. and Hauser, K. (2013). Artificial intelligence framework for simulating clinical decision-making: A markov decision process approach. Artificial intelligence in medicine, 57(1):9–19.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb):281–305.

[Bergstra et al., 2013] Bergstra, J., Yamins, D., and Cox, D. D. (2013). Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In Proceedings of the 12th Python in science conference, pages 13–20. Citeseer.

[Billis and Bamidis, 2014] Billis, A. and Bamidis, P. D. (2014). Employing time-series forecasting to historical medical data: an application towards early prognosis within elderly health monitoring environments. In AI-AM/NetMed@ ECAI, pages 31–35. Citeseer.

[Bishop, 2006] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

[Bonaccorso, 2017] Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.

[Box et al., 2015] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

[Breiman, 2001] Breiman, L. (2001). Random forests. Machine learning, 45(1):5–32.

[Cao et al., 2012] Cao, Q., Ewing, B. T., and Thompson, M. A. (2012). Forecasting wind speed with recurrent neural networks. European Journal of Operational Research, 221(1):148–154.

[Carlin et al., 2018] Carlin, C. S., Ho, L. V., Ledbetter, D. R., Aczon, M. D., and Wetzel, R. C. (2018). Predicting individual physiologically acceptable states at discharge from a pediatric intensive care unit. Journal of the American Medical Informatics Association, 25(12):1600–1607.

[Chauhan and Vig, 2015] Chauhan, S. and Vig, L. (2015). Anomaly detection in ecg time signals via deep long short-term memory networks. In 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pages 1–7. IEEE.

[Che et al., 2015] Che, Z., Kale, D., Li, W., Bahadori, M. T., and Liu, Y. (2015). Deep computational phenotyping. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 507–516. ACM.

[Chen et al., 2009] Chen, X., Xu, D., Zhang, G., and Mukkamala, R. (2009). Forecasting acute hypotensive episodes in intensive care patients based on a peripheral arterial blood pressure waveform. In 2009 36th Annual Computers in Cardiology Conference (CinC), pages 545–548. IEEE.

[Cheung et al., 2015] Cheung, C. C., Martyn, A., Campbell, N., Frost, S., Gilbert, K., Michota, F., Seal, D., Ghali, W., and Khan, N. A. (2015). Predictors of intraoperative hypotension and bradycardia. The American journal of medicine, 128(5):532–538.

[Chobanian et al., 2003] Chobanian, A. V., Bakris, G. L., Black, H. R., Cushman, W. C., Green, L. A., Izzo Jr, J. L., Jones, D. W., Materson, B. J., Oparil, S., Wright Jr, J. T., et al. (2003). The seventh report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure: the jnc 7 report. Jama, 289(19):2560–2571.

[Chollet, 2018] Chollet, F. (2018). Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG.

[Chollet et al., 2015] Chollet, F. et al. (2015). Keras, github repository, 2015.

[Churpek et al., 2016] Churpek, M. M., Yuen, T. C., Winslow, C., Meltzer, D. O., Kattan, M. W., and Edelson, D. P. (2016). Multicenter comparison of machine learning methods and conventional regression for predicting clinical deterioration on the wards. Critical care medicine, 44(2):368.

[Clifford et al., 2008] Clifford, G. D., Long, W., Moody, G., and Szolovits, P. (2008). Robust parameter extraction for decision support using multimodal intensive care data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 367(1887):411–429.

[Clifton et al., 2015] Clifton, D., Niehaus, K., Charlton, P., and Colopy, G. (2015). Health informatics via machine learning for the clinical management of patients. Yearbook of medical informatics, 24(01):38–43.

[Cook, 2008] Cook, N. R. (2008). Statistical evaluation of prognostic versus diagnostic models: beyond the roc curve. Clinical chemistry, 54(1):17–23.

[Crespo et al., 2002] Crespo, C., McNames, J., Aboy, M., Bassale, J., Ellenby, M., Lai, S., and Goldstein, B. (2002). Precursors in the arterial blood pressure signal to episodes of acute hypotension in sepsis. In Proceedings of the 16th international EURASIP Conference BIOSIGNAL, volume 16, pages 206–208.

[Deng and Jirutitijaroen, 2010] Deng, J. and Jirutitijaroen, P. (2010). Short-term load forecasting using time series analysis: A case study for singapore. In 2010 IEEE Conference on Cybernetics and Intelligent Systems, pages 231–236. IEEE.

[Dietterich, 2002] Dietterich, T. G. (2002). Machine learning for sequential data: A review. In Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR), pages 15–30. Springer.

[Donald et al., 2012] Donald, R., Howells, T., Piper, I., Chambers, I., Citerio, G., Enblad, P., Gregson, B., Kiening, K., Mattern, J., Nilsson, P., et al. (2012). Early warning of eusig-defined hypotensive events using a bayesian artificial neural network. In Intracranial Pressure and Brain Monitoring XIV, pages 39–44. Springer.

[Du Preez and Witt, 2003] Du Preez, J. and Witt, S. F. (2003). Univariate versus multivariate time series forecasting: an application to international tourism demand. International Journal of Forecasting, 19(3):435–451.

[Elfadil and Ibrahim, 2011] Elfadil, N. and Ibrahim, I. (2011). Self organizing neural network approach for identification of patients with congestive heart failure. In 2011 International Conference on Multimedia Computing and Systems, pages 1–6. IEEE.

[Elliott and Coventry, 2012] Elliott, M. and Coventry, A. (2012). Critical care: the eight vital signs of patient monitoring. British Journal of Nursing, 21(10):621–625.

[Eshelman et al., 2008] Eshelman, L. J., Lee, K., Frassica, J. J., Zong, W., Nielsen, L., and Saeed, M. (2008). Development and evaluation of predictive alerts for hemodynamic instability in icu patients. In AMIA Annual Symposium Proceedings, volume 2008, page 379. American Medical Informatics Association.

[Eswaran et al., 2010] Eswaran, C., Logeswaran, R., et al. (2010). An adaptive hybrid algorithm for time series prediction in healthcare. In 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, pages 21–26. IEEE.

[Forkan et al., 2017] Forkan, A. R. M., Khalil, I., and Atiquzzaman, M. (2017). Visibid: A learning model for early discovery and real-time prediction of severe clinical events using vital signs as big data. Computer Networks, 113:244–257.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, pages 1189–1232.

[Frizzell et al., 2017] Frizzell, J. D., Liang, L., Schulte, P. J., Yancy, C. W., Heidenreich, P. A., Hernandez, A. F., Bhatt, D. L., Fonarow, G. C., and Laskey, W. K. (2017). Prediction of 30-day all-cause readmissions in patients hospitalized for heart failure: comparison of machine learning and other statistical approaches. Jama cardiology, 2(2):204–209.

[Fu et al., 2016] Fu, R., Zhang, Z., and Li, L. (2016). Using lstm and gru neural network methods for traffic flow prediction. In 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pages 324–328. IEEE.

[Funahashi, 1989] Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. Neural networks, 2(3):183–192.

[Gather et al., 2002] Gather, U., Imhoff, M., and Fried, R. (2002). Graphical models for multivariate time series from intensive care monitoring. Statistics in medicine, 21(18):2685–2701.

[Gensler et al., 2016] Gensler, A., Henze, J., Sick, B., and Raabe, N. (2016). Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks. In 2016 IEEE international conference on systems, man, and cybernetics (SMC), pages 002858–002865. IEEE.

[Ghaffari et al., 2010] Ghaffari, A., Homaeinezhad, M., Atarod, M., and Akraminia, M. (2010). A methodology for prediction of acute hypotensive episodes in icu via a risk scoring model including analysis of st-segment variations. Cardiovascular Engineering, 10(1):12–29.

[Ghassemi et al., 2015] Ghassemi, M., Pimentel, M. A., Naumann, T., Brennan, T., Clifton, D. A., Szolovits, P., and Feng, M. (2015). A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In Twenty-Ninth AAAI Conference on Artificial Intelligence.

[Ghosh et al., 2016] Ghosh, S., Feng, M., Nguyen, H., and Li, J. (2016). Hypotension risk prediction via sequential contrast patterns of icu blood pressure. IEEE journal of biomedical and health informatics, 20(5):1416–1426.

[Ghosh et al., 2017] Ghosh, S., Li, J., Cao, L., and Ramamohanarao, K. (2017). Septic shock prediction for icu patients via coupled hmm walking on sequential contrast patterns. Journal of biomedical informatics, 66:19–31.

[Gollapudi, 2016] Gollapudi, S. (2016). Practical machine learning. Packt Publishing Ltd.

[Golubnitschaja et al., 2014] Golubnitschaja, O., Kinkorova, J., and Costigliola, V. (2014). Predictive, preventive and personalised medicine as the hardcore of 'horizon 2020': Epma position paper. EPMA Journal, 5(1):6.

[Graves, 2012] Graves, A. (2012). Supervised sequence labelling with recurrent neural networks. 2012. URL http://books. google. com/books.

[Graves, 2013] Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

[Graves et al., 2013] Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 6645–6649. IEEE.

[Guidi et al., 2014] Guidi, G., Pettenati, M. C., Melillo, P., and Iadanza, E. (2014). A machine learning system to improve heart failure patient assistance. IEEE journal of biomedical and health informatics, 18(6):1750–1756.

[Guidi et al., 2015] Guidi, G., Pollonini, L., Dacso, C. C., and Iadanza, E. (2015). A multi-layer monitoring system for clinical management of congestive heart failure. BMC medical informatics and decision making, 15(3):S5.

[Gulshan et al., 2016] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama, 316(22):2402–2410.

[Gupta and Wang, 2010] Gupta, S. and Wang, L. (2010). Stock forecasting with feedforward neural networks and gradual data sub-sampling. Australian Journal of Intelligent Information Processing Systems, 11(4):14–17.

[Hatib et al., 2018] Hatib, F., Jian, Z., Buddi, S., Lee, C., Settels, J., Sibert, K., Rinehart, J., and Cannesson, M. (2018). Machine-learning algorithm to predict hypotension based on high-fidelity arterial pressure waveform analysis. Anesthesiology: The Journal of the American Society of Anesthesiologists, 129(4):663–674.

[Hill et al., 1996] Hill, T., O'Connor, M., and Remus, W. (1996). Neural network models for time series forecasts. Management science, 42(7):1082–1092.

[Ho et al., 2002] Ho, S., Xie, M., and Goh, T. (2002). A comparative study of neural network and box-jenkins arima modeling in time series prediction. Computers & Industrial Engineering, 42(2-4):371–375.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8):1735–1780.

[Huang and Kuo, 2018] Huang, C.-J. and Kuo, P.-H. (2018). A deep cnn-lstm model for particulate matter (pm2. 5) forecasting in smart cities. Sensors, 18(7):2220.

[Hui et al., 2017] Hui, C., Cardinale, M., and Yegneswaran, B. (2017). Significant bradycardia in critically ill patients receiving dexmedetomidine and fentanyl. Case reports in critical care, 2017.

[Ince et al., 2016] Ince, T., Kiranyaz, S., Eren, L., Askar, M., and Gabbouj, M. (2016). Real-time motor fault detection by 1-d convolutional neural networks. IEEE Transactions on Industrial Electronics, 63(11):7067–7075.

[Irwin and Rippe, 2008] Irwin, R. S. and Rippe, J. M. (2008). Irwin and Rippe's intensive care medicine. Lippincott Williams & Wilkins.

[Isler, 2016] Isler, Y. (2016). Discrimination of systolic and diastolic dysfunctions using multi-layer perceptron in heart rate variability analysis. Computers in biology and medicine, 76:113–119.

[Jain and Medsker, 2000] Jain, L. C. and Medsker, L. R. (2000). Recurrent neural networks: design and applications. CRC Press.

[Jo et al., 2017] Jo, Y., Lee, L., and Palaskar, S. (2017). Combining lstm and latent topic modeling for mortality prediction. arXiv preprint arXiv:1709.02842.

[Johnson et al., 2016a] Johnson, A. E., Ghassemi, M. M., Nemati, S., Niehaus, K. E., Clifton, D. A., and Clifford, G. D. (2016a). Machine learning and decision support in critical care. Proceedings of the IEEE. Institute of Electrical and Electronics Engineers, 104(2):444.

[Johnson et al., 2017] Johnson, A. E., Pollard, T. J., and Mark, R. G. (2017). Reproducibility in critical care: a mortality prediction case study. In Machine Learning for Healthcare Conference, pages 361–376.

[Johnson et al., 2016b] Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016b). Mimic-iii, a freely accessible critical care database. Scientific data, 3:160035.

[Jovic and Bogunovic, 2011] Jovic, A. and Bogunovic, N. (2011). Electrocardiogram analysis using a combination of statistical, geometric, and nonlinear heart rate variability features. Artificial intelligence in medicine, 51(3):175–186.

[Kam, 2014] Kam, K. M. (2014). Stationary and non-stationary time series prediction using state space model and pattern-based approach. The University of Texas at Arlington.

[Kam, 2015] Kam, K. M. (2015). Stationary and non-stationary time series prediction using state space model and pattern-based approach.

[Khashei and Bijari, 2011] Khashei, M. and Bijari, M. (2011). A novel hybridization of artificial neural networks and arima models for time series forecasting. Applied Soft Computing, 11(2):2664–2675.

[Khashei et al., 2012] Khashei, M., Bijari, M., and Hejazi, S. R. (2012). Combining seasonal arima models with computational intelligence techniques for time series forecasting. Soft Computing, 16(6):1091–1105.

[Kim et al., 2019] Kim, M., Choi, W., Jeon, Y., and Liu, L. (2019). A hybrid neural network model for power demand forecasting. Energies, 12(5):931.

[Kim and Kim, 2019] Kim, T. and Kim, H. Y. (2019). Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. PloS one, 14(2):e0212320.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[Kiranyaz et al., 2016] Kiranyaz, S., Ince, T., and Gabbouj, M. (2016). Real-time patient-specific ecg classification by 1-d convolutional neural networks. IEEE Transactions on Biomedical Engineering, 63(3):664–675.

[Klabunde, 2011] Klabunde, R. (2011). Cardiovascular physiology concepts. Lippincott Williams & Wilkins.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

[Kusumoto et al., 2018] Kusumoto, F. M., Schoenfeld, M. H., Barrett, C., Edgerton, J. R., Ellenbogen, K. A., Gold, M. R., Goldschlager, N. F., Hamilton, R. M., Joglar, J. A., Kim, R. J., et al. (2018). 2018 acc/aha/hrs guideline on the evaluation and management of patients with bradycardia and cardiac conduction delay: a report of the american college of cardiology/american heart association task force on clinical practice guidelines and the heart rhythm society. Journal of the American College of Cardiology, page 25701.

[Kwon et al., 2018] Kwon, J.-m., Lee, Y., Lee, Y., Lee, S., and Park, J. (2018). An algorithm based on deep learning for predicting in-hospital cardiac arrest. Journal of the American Heart Association, 7(13):e008678.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[Langley et al., 2009] Langley, P., King, S., Zheng, D., Bowers, E., Wang, K., Allen, J., and Murray, A. (2009). Predicting acute hypotensive episodes from mean arterial pressure. In 2009 36th Annual Computers in Cardiology Conference (CinC), pages 553–556. IEEE.

[Lasko et al., 2013] Lasko, T. A., Denny, J. C., and Levy, M. A. (2013). Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. PloS one, 8(6):e66341.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. nature, 521(7553):436.

[Lee and Mark, 2010a] Lee, J. and Mark, R. (2010a). A hypotensive episode predictor for intensive care based on heart rate and blood pressure time series. In 2010 Computing in Cardiology, pages 81–84. IEEE.

[Lee and Mark, 2010b] Lee, J. and Mark, R. G. (2010b). An investigation of patterns in hemodynamic data indicative of impending hypotension in intensive care. Biomedical engineering online, 9(1):62.

[Lesser et al., 2003] Lesser, J. B., Sanborn, K. V., Valskys, R., and Kuroda, M. (2003). Severe bradycardia during spinal and epidural anesthesia recorded by an anesthesia information management system. Anesthesiology: The Journal of the American Society of Anesthesiologists, 99(4):859–866.

[Li and Hu, 2012] Li, C. and Hu, J.-W. (2012). A new arima-based neuro-fuzzy approach and swarm intelligence for time series forecasting. Engineering Applications of Artificial Intelligence, 25(2):295–308.

[Li et al., 2017] Li, X., Wu, S., and Wang, L. (2017). Blood pressure prediction via recurrent models with contextual layer. In Proceedings of the 26th International Conference on World Wide Web, pages 685–693. International World Wide Web Conferences Steering Committee.

[Li-wei et al., 2014] Li-wei, H. L., Nemati, S., Moody, G. B., Heldt, T., and Mark, R. G. (2014). Uncovering clinical significance of vital sign dynamics in critical care. In Computing in Cardiology 2014, pages 1141–1144. IEEE.

[Liang and Srikant, 2016] Liang, S. and Srikant, R. (2016). Why deep neural networks for function approximation? arXiv preprint arXiv:1610.04161.

[Lineesh et al., 2010] Lineesh, M., Minu, K., and John, C. J. (2010). Analysis of non-stationary nonlinear economic time series of gold price: A comparative study. In International Mathematical Forum, volume 5, pages 1673–1683. Citeseer.

[Lipton et al., 2015a] Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015a). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

[Lipton et al., 2015b] Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. (2015b). Learning to diagnose with lstm recurrent neural networks. arXiv preprint arXiv:1511.03677.

[Luong et al., 2015] Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2015). Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114.

[Ma et al., 2015] Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies, 54:187–197.

[Mandal et al., 2007] Mandal, P., Senjyu, T., Urasaki, N., Yona, A., Funabashi, T., and Srivastava, A. K. (2007). Price forecasting for day-ahead electricity market using recursive neural network. In 2007 IEEE Power Engineering Society General Meeting, pages 1–8. IEEE.

[Mangrum and DiMarco, 2000] Mangrum, J. M. and DiMarco, J. P. (2000). The evaluation and management of bradycardia. New England Journal of Medicine, 342(10):703–709.

[Mani et al., 2014] Mani, S., Ozdas, A., Aliferis, C., Varol, H. A., Chen, Q., Carnevale, R., Chen, Y., Romano-Keeler, J., Nian, H., and Weitkamp, J.-H. (2014). Medical decision support using machine learning for early detection of late-onset neonatal sepsis. Journal of the American Medical Informatics Association, 21(2):326–336.

[Mann et al., 2014] Mann, D. L., Zipes, D. P., Libby, P., and Bonow, R. O. (2014). Braunwald's Heart Disease E-Book: A Textbook of Cardiovascular Medicine. Elsevier Health Sciences.

[Marlin et al., 2012] Marlin, B. M., Kale, D. C., Khemani, R. G., and Wetzel, R. C. (2012). Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In Proceedings of the 2nd ACM SIGHIT international health informatics symposium, pages 389–398. ACM.

[Masetic and Subasi, 2016] Masetic, Z. and Subasi, A. (2016). Congestive heart failure detection using random forest classifier. Computer methods and programs in biomedicine, 130:54–64.

[Masum et al., 2017] Masum, S., Liu, Y., and Chiverton, J. (2017). Comparative analysis of the outcomes of differing time series forecasting strategies. In 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pages 1964–1968. IEEE.

[Masum et al., 2018] Masum, S., Liu, Y., and Chiverton, J. (2018). Multi-step time series forecasting of electric load using machine learning models. In International Conference on Artificial Intelligence and Soft Computing, pages 148–159. Springer.

[McGlynn et al., 2003] McGlynn, E. A., Asch, S. M., Adams, J., Keesey, J., Hicks, J., DeCristofaro, A., and Kerr, E. A. (2003). The quality of health care delivered to adults in the united states. New England journal of medicine, 348(26):2635–2645.

[McKinney, 2012] McKinney, W. (2012). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.".

[Mclachlan et al., 2007] Mclachlan, K., Jenkins, A., and O'NEAL, D. (2007). The role of continuous glucose monitoring in clinical decision-making in diabetes in pregnancy. Australian and New Zealand journal of obstetrics and gynaecology, 47(3):186–190.

[Meyler et al., 1998] Meyler, A., Kenny, G., and Quinn, T. (1998). Forecasting irish inflation using arima models.

[Mitchell, 1997] Mitchell, T. M. (1997). Artificial neural networks. Machine learning, 45:81–127.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G.,

et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540):529.

[Mohamed et al., 2012] Mohamed, A.-r., Hinton, G., and Penn, G. (2012). Understanding how deep belief networks perform acoustic modelling. neural networks, pages 6–9.

[Molaei and Keyvanpour, 2015] Molaei, S. M. and Keyvanpour, M. R. (2015). An analytical review for event prediction system on time series. In 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), pages 1–6. IEEE.

[Moody and Lehman, 2009] Moody, G. B. and Lehman, L.-w. H. (2009). Predicting acute hypotensive episodes: The 10th annual physionet/computers in cardiology challenge. In 2009 36th Annual Computers in Cardiology Conference (CinC), pages 541–544. IEEE.

[Müller et al., 2016] Müller, A. C., Guido, S., et al. (2016). Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

[Murphy, 2012] Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

[Narin et al., 2014] Narin, A., Isler, Y., and Ozer, M. (2014). Investigating the performance improvement of hrv indices in chf using feature selection methods based on backward elimination and statistical significance. Computers in biology and medicine, 45:72–79.

[Nguyen et al., 2017] Nguyen, P., Tran, T., and Venkatesh, S. (2017). Deep learning to attend to risk in icu. arXiv preprint arXiv:1707.05010.

[Olaiya and Adeyemo, 2012] Olaiya, F. and Adeyemo, A. B. (2012). Application of data mining techniques in weather prediction and climate change studies. International Journal of Information Engineering and Electronic Business, 4(1):51.

[Ongenae et al., 2013] Ongenae, F., Van Looy, S., Verstraeten, D., Verplancke, T., Benoit, D., De Turck, F., Dhaene, T., Schrauwen, B., and Decruyenaere, J. (2013). Time series classification for the prediction of dialysis in critically ill patients using

echo statenetworks. Engineering Applications of Artificial Intelligence, 26(3):984–996.

[Oresko et al., 2010] Oresko, J. J., Jin, Z., Cheng, J., Huang, S., Sun, Y., Duschl, H., and Cheng, A. C. (2010). A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. IEEE Transactions on Information Technology in Biomedicine, 14(3):734–740.

[Panahiazar et al., 2015] Panahiazar, M., Taslimitehrani, V., Pereira, N., and Pathak, J. (2015). Using ehrs and machine learning for heart failure survival analysis. Studies in health technology and informatics, 216:40.

[Pascanu et al., 2013a] Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2013a). How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026.

[Pascanu et al., 2013b] Pascanu, R., Mikolov, T., and Bengio, Y. (2013b). On the difficulty of training recurrent neural networks. In International conference on machine learning, pages 1310–1318.

[Patnaik et al., 2011] Patnaik, D., Butler, P., Ramakrishnan, N., Parida, L., Keller, B. J., and Hanauer, D. A. (2011). Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 360–368. ACM.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct):2825–2830.

[Perer and Wang, 2014] Perer, A. and Wang, F. (2014). Frequence: interactive mining and visualization of temporal frequent event sequences. In Proceedings of the 19th international conference on Intelligent User Interfaces, pages 153–162. ACM.

[Pinsky, 2007] Pinsky, M. R. (2007). Hemodynamic evaluation and monitoring in the icu. Chest, 132(6):2020–2029.

[Pocock et al., 2005] Pocock, S. J., Wang, D., Pfeffer, M. A., Yusuf, S., McMurray, J. J., Swedberg, K. B., Ostergren, J., Michelson, E. L., Pieper, K. S., and Granger, C. B. (2005). Predictors of mortality and morbidity in patients with chronic heart failure. European heart journal, 27(1):65–75.

[Pollard, 2001] Pollard, J. B. (2001). Cardiac arrest during spinal anesthesia: common mechanisms and strategies for prevention. Anesthesia & Analgesia, 92(1):252–256.

[Quinlan, 2014] Quinlan, J. R. (2014). C4. 5: programs for machine learning. Elsevier.

[Radzuan et al., 2013] Radzuan, N. F. M., Othman, Z., and Bakar, A. A. (2013). Uncertain time series in weather prediction. Procedia Technology, 11:557–564.

[Raschka, 2015] Raschka, S. (2015). Python machine learning. Packt Publishing Ltd.

[Ren et al., 2016] Ren, Y., Suganthan, P. N., Srikanth, N., and Amaratunga, G. (2016). Random vector functional link network for short-term electricity load demand forecasting. Information Sciences, 367:1078–1093.

[Rendle, 2010] Rendle, S. (2010). Factorization machines. In 2010 IEEE International Conference on Data Mining, pages 995–1000. IEEE.

[Rocha et al., 2011] Rocha, T., Paredes, S., de Carvalho, P., and Henriques, J. (2011). Prediction of acute hypotensive episodes by means of neural network multi-models. Computers in Biology and Medicine, 41(10):881–890.

[Saeed et al., 2011] Saeed, M., Villarroel, M., Reisner, A. T., Clifford, G., Lehman, L.-W., Moody, G., Heldt, T., Kyaw, T. H., Moody, B., and Mark, R. G. (2011). Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. Critical care medicine, 39(5):952.

[Samuel, 1988] Samuel, A. L. (1988). Some studies in machine learning using the game of checkers. ii—recent progress. In Computer Games I, pages 366–400. Springer.

[Schaefer et al., 2005] Schaefer, A. J., Bailey, M. D., Shechter, S. M., and Roberts, M. S. (2005). Modeling medical treatment using markov decision processes. In Operations research and health care, pages 593–612. Springer.

[Schuster and Paliwal, 1997] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681.

[Shashikumar et al., 2017] Shashikumar, S. P., Stanley, M. D., Sadiq, I., Li, Q., Holder, A., Clifford, G. D., and Nemati, S. (2017). Early sepsis detection in critical care patients using multiscale blood pressure and heart rate dynamics. Journal of electrocardiology, 50(6):739–743.

[Shumway and Stoffer, 2017] Shumway, R. H. and Stoffer, D. S. (2017). Time series analysis and its applications: with R examples. Springer.

[Sideris et al., 2016] Sideris, C., Kalantarian, H., Nemati, E., and Sarrafzadeh, M. (2016). Building continuous arterial blood pressure prediction models using recurrent networks. In 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pages 1–5. IEEE.

[Smola and Schölkopf, 2004] Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and computing, 14(3):199–222.

[Son et al., 2012] Son, C.-S., Kim, Y.-N., Kim, H.-S., Park, H.-S., and Kim, M.-S. (2012). Decision-making model for early diagnosis of congestive heart failure using rough set and decision tree approaches. Journal of biomedical informatics, 45(5):999–1008.

[Su et al., 2018] Su, P., Ding, X.-R., Zhang, Y.-T., Liu, J., Miao, F., and Zhao, N. (2018). Long-term blood pressure prediction with deep recurrent neural networks. In 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), pages 323–328. IEEE.

[Subbe et al., 2001] Subbe, C., Kruger, M., Rutherford, P., and Gemmel, L. (2001). Validation of a modified early warning score in medical admissions. Qjm, 94(10):521–526.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112.

[Tabachnick et al., 2007] Tabachnick, B. G., Fidell, L. S., and Ullman, J. B. (2007). Using multivariate statistics, volume 5. Pearson Boston, MA.

[Taieb et al., 2012] Taieb, S. B., Bontempi, G., Atiya, A. F., and Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications, 39(8):7067–7083.

[Tian and Pan, 2015] Tian, Y. and Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity), pages 153–158. IEEE.

[Trevor et al., 2009] Trevor, H., Robert, T., and JH, F. (2009). The elements of statistical learning: data mining, inference, and prediction.

[van der Heijden et al., 2014] van der Heijden, M., Velikova, M., and Lucas, P. J. (2014). Learning bayesian networks for clinical time series analysis. Journal of biomedical informatics, 48:94–105.

[Vedomske et al., 2013] Vedomske, M. A., Brown, D. E., and Harrison, J. H. (2013). Random forests on ubiquitous data for heart failure 30-day readmissions prediction. In 2013 12th International Conference on Machine Learning and Applications, volume 2, pages 415–421. IEEE.

[Vespa et al., 1999] Vespa, P. M., Nenov, V., and Nuwer, M. R. (1999). Continuous eeg monitoring in the intensive care unit: early findings and clinical efficacy. Journal of Clinical Neurophysiology, 16(1):1–13.

[Walsh et al., 2013] Walsh, M., Devereaux, P. J., Garg, A. X., Kurz, A., Turan, A., Rodseth, R. N., Cywinski, J., Thabane, L., and Sessler, D. I. (2013). Relationship between intraoperative mean arterial pressure and clinical outcomes after noncardiac surgerytoward an empirical definition of hypotension. Anesthesiology: The Journal of the American Society of Anesthesiologists, 119(3):507–515.

[Weigend, 2018] Weigend, A. S. (2018). Time series prediction: forecasting the future and understanding the past. Routledge.

[Witten et al., 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[Wu et al., 2010] Wu, J., Roy, J., and Stewart, W. F. (2010). Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches. Medical care, pages S106–S113.

[Wu et al., 2017] Wu, M., Ghassemi, M., Feng, M., Celi, L. A., Szolovits, P., and Doshi-Velez, F. (2017). Understanding vasopressor intervention and weaning: Risk prediction in a public heterogeneous clinical time series database. Journal of the American Medical Informatics Association, 24(3):488–495.

[Xue et al., 2018] Xue, Y., Klabjan, D., and Luo, Y. (2018). Predicting icu readmission using grouped physiological and medication trends. Artificial intelligence in medicine.

[Yamins and DiCarlo, 2016] Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. Nature neuroscience, 19(3):356.

[Yong et al., 2015] Yong, J., Hibbert, P., Runciman, W. B., and Coventry, B. J. (2015). Bradycardia as an early warning sign for cardiac arrest during routine laparoscopic surgery. International Journal for Quality in Health Care, 27(6):473–478.

[Yorozu et al., 2007] Yorozu, T., Iijima, T., Matsumoto, M., Yeo, X., and Takagi, T. (2007). Factors influencing intraoperative bradycardia in adult patients. Journal of anesthesia, 21(2):136–141.

[Zhai et al., 2014] Zhai, H., Brady, P., Li, Q., Lingren, T., Ni, Y., Wheeler, D. S., and Solti, I. (2014). Developing and evaluating a machine learning based algorithm to predict the need of pediatric intensive care unit transfer for newly hospitalized children. Resuscitation, 85(8):1065–1071.

[Zhou et al., 2016] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 207–212.

[Zhu and Wang, 2010] Zhu, M. and Wang, L. (2010). Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms. In The 2010 International Joint Conference on Neural Networks (IJCNN), pages 1–5. IEEE.

[Zhu et al., 2018] Zhu, Y., Fan, X., Wu, J., Liu, X., Shi, J., and Wang, C. (2018). Predicting icu mortality by supervised bidirectional lstm networks. In AIH@IJCAI.

[Zolfaghar et al., 2013] Zolfaghar, K., Meadem, N., Teredesai, A., Roy, S. B., Chin, S.-C., and Muckian, B. (2013). Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In 2013 IEEE International Conference on Big Data, pages 64–71. IEEE.

Appendices

Appendix A

Certificate of Ethical Review

Certificate of Ethics Review

Project Title:	Machine Learning Scopes in Time Series Forecasting of Physiological Data
User ID:	642495
Name:	Shamsul Masum
Application Date:	09/08/2018 15:07:03

You must download your certificate, print a copy and keep it as a record of this review.

It is your responsibility to adhere to the University Ethics Policy and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers and University Health and Safety Policy.

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

You are reminded that as a University of Portsmouth Researcher you are bound by the UKRIO Code of Practice for Research; any breach of this code could lead to action being taken following the University's Procedure for the Investigation of Allegations of Misconduct in Research.

Any changes in the answers to the questions reflecting the design, management or conduct of the research over the course of the project must be notified to the Faculty Ethics Committee. Any changes that affect the answers given in the questionnaire, not reported to the Faculty Ethics Committee, will invalidate this certificate.

This ethical review should not be used to infer any comment on the academic merits or methodology of the project. If you have not already done so, you are advised to develop a clear protocol/proposal and ensure that it is independently reviewed by peers or others of appropriate standing. A favourable ethical opinion should not be perceived as permission to proceed with the research; there might be other matters of governance which require further consideration including the agreement of any organisation hosting the research.

GovernanceChecklist
A1-BriefDescriptionOfProject: A patient’s physiological status can deteriorate suddenly without any prior notification. When this happens, the current healthcare paradigm requires that a healthcare professional act swiftly to save the patient. This paradigm can be referred to as a “reactive patient care paradigm”. Existing early warning systems consist of predefined rules which are applied to vital signs to generate a reactive alarm and in some cases generate a significant number of false alarms. Researchers have found that there have been many cases where patients have received an incorrect diagnosis and treatment. Furthermore, there is a huge gap between research and clinical

practice. The European Association for Predictive, Preventive and Personalised Medicine (EPMA) has criticised the current health care model around Europe and worldwide. EPMA has motivated its representatives to explore advanced research in the healthcare area in order to come up with a solution to overcome these types of scenarios.

Vital signs of the human body reflect the physiological state of the patient. Researchers have identified that psychological functions also known as vital signs such as temperature, blood pressure, heart rate, oxygen saturation, electrocardiography, respiratory rate and pulse oximetry are generally subject to measure to diagnose an individual health condition. Advancement in medical technology is allowing us to capture vital sign data from patients. Intensive Care Units (ICU) at hospitals are considered the best place for a wealth of patient's data collected from many advanced specialised units. These captured data then could be stored in a database which could be a source of much valuable information in the form of important events, patterns, image, scan, lab results, waveforms and much more. With this wealth of patients' data, then a question should raise as to whether it is possible to predict the future of a patient's physiological state. Forecasting on a patient's physiological state could then alert a healthcare professional about the patient who is in imminent threat of physiological deterioration or for another patient that is going to be stable for the next few hours and then suddenly deteriorate. Such scenarios can help a healthcare professional to intervene in advance according to the situation and thus contribute to a change in the reactive patient care paradigm to help provide a predictive patient care. The challenges in terms of achieving this predictive patient care paradigm are access to a rich enough source of data, to then understand these data, and to learn from these data. Furthermore, it is important to fully explore events such as deterioration events and then look back from that event to find whether there was any signature left to identify it. Part of this challenge is to build an algorithm that can identify and capture that signature. Such an algorithm can then be deployed to the patient's bedside for further testing to see whether the developed algorithm is able to identify patients in advance who might be at risk of a serious deterioration in their physiological state.

Time series forecasting has been used in economics and statistics, where linear statistical models such as ARIMA has been used to predict linear data. However, the most real-world application involves nonlinear data, yet nonlinear time series forecast and analysis lag behind compared to the development linear time series. Machine Learning found to be the most active tools in modelling the forecast model in recent times. Machine learning is a field which helps to learn from data by developing algorithms. Developed Machine learning algorithms have produced superior performance in applications like natural language processing, speech recognition and spam detection etc. Machine learning models are also known as data-driven models which deal with nonlinear data by learning the stochastic dependency between a set of input and output variables of the past data. However, there has been very little research at the intersection of time series forecasting and machine learning. The scope of Machine learning algorithms and methodologies have received little attention in long-term or multi-step ahead forecast. Few literatures have considered machine learning algorithms which have been limited to single step forecast. Whereas multi-step ahead would be great and considered to be a challenging task. The gaps between theory and practice of forecasting need to be minimised. In practice of forecasting task like data cleansing, feature engineering and balance between computational complexity and statistical accuracy are often neglected in theory of forecast.

Keeping all that in mind the aim of the thesis was to investigate the scope of machine learning in multi-step ahead forecasting of univariate and multivariate vital sign time series data. vital sign time series data has been extracted from the MIMIC database. MIMIC is an openly available dataset developed by the MIT Lab for Computational Physiology, comprising deidentified health data associated with 40,000 critical care patients. It includes demographics, vital signs, laboratory tests, medications, and more.

A2-Faculty: Technology
A3-VoluntarilyReferToFEC: No
A5-AlreadyExternallyReviewed: No
B1-HumanParticipants: No
HumanParticipantsDefinition
B2-HumanParticipantsConfirmation: Yes
B4-InvolvesNHSPatients: No
B5-NoConsentOrDeception: No
B7-InvolvesUninformedOrDependents: No
B9-FinancialInducements: No
C1-DrugsPlacebosOrOtherSubstances: No
C2-BloodOrTissueSamples: No
C3-PainOrMildDiscomfort: No
C4-PsychologicalStressOrAnxiety: No
C5-ProlongedOrRepetitiveTesting: No
C6-SafetyRisksBeyondAssessment: No
D2-PhysicalEcologicalDamage: No
PhysicalEcologicalDamageWarning
D4-HistoricalOrCulturalDamage: No
HistoricalOrCulturalDamageWarning
E1-ContentiousOrIllegal: No
ContentiousOrIllegalWarning
E2-SociallySensitiveIssues: No
SociallySensitiveWarning
F1-InvolvesAnimals: No
InvolvesAnimalsWarning
F2-HarmfulToThirdParties: No
HarmfulToThirdPartiesWarning
G1-ConfirmReadEthicsPolicy: Confirmed
G2-ConfirmReadUKRIOCodeOfPractice: Confirmed
G3-ConfirmReadConcordatToSupportResearchIntegrity: Confirmed
G4-ConfirmedCorrectInformation: Confirmed

Appendix B

Collaborative Institutional Training Initiative

COLLABORATIVE INSTITUTIONAL TRAINING INITIATIVE (CITI PROGRAM)

COMPLETION REPORT - PART 1 OF 2
COURSEWORK REQUIREMENTS*

* NOTE: Scores on this Requirements Report reflect quiz completions at the time all requirements for the course were met. See list below for details. See separate Transcript Report for more recent quiz scores, including those on optional (supplemental) course elements.

- **Name:** shamsul masum (ID: 6207588)
- **Institution Affiliation:** Massachusetts Institute of Technology Affiliates (ID: 1912)
- **Institution Email:** up642495@myport.ac.uk
- **Institution Unit:** school of engineering

- **Curriculum Group:** Human Research
- **Course Learner Group:** Data or Specimens Only Research
- **Stage:** Stage 1 - Basic Course

- **Record ID:** 22496305
- **Completion Date:** 04-Mar-2017
- **Expiration Date:** 03-Mar-2020
- **Minimum Passing:** 90
- **Reported Score*:** 92

REQUIRED AND ELECTIVE MODULES ONLY	DATE COMPLETED	SCORE
Belmont Report and CITI Course Introduction (ID: 1127)	02-Mar-2017	3/3 (100%)
History and Ethics of Human Subjects Research (ID: 498)	02-Mar-2017	6/7 (86%)
Basic Institutional Review Board (IRB) Regulations and Review Process (ID: 2)	02-Mar-2017	5/5 (100%)
Records-Based Research (ID: 5)	02-Mar-2017	3/3 (100%)
Genetic Research in Human Populations (ID: 6)	02-Mar-2017	4/5 (80%)
Populations in Research Requiring Additional Considerations and/or Protections (ID: 16680)	04-Mar-2017	5/5 (100%)
Research and HIPAA Privacy Protections (ID: 14)	04-Mar-2017	4/5 (80%)
Conflicts of Interest in Research Involving Human Subjects (ID: 488)	04-Mar-2017	5/5 (100%)
Massachusetts Institute of Technology (ID: 1290)	04-Mar-2017	No Quiz

For this Report to be valid, the learner identified above must have had a valid affiliation with the CITI Program subscribing institution identified above or have been a paid Independent Learner.

Verify at: www.citiprogram.org/verify/?k3275c93f-e1f9-40fa-a3a5-47f6647ca7bb-22496305

Collaborative Institutional Training Initiative (CITI Program)
Email: support@citiprogram.org
Phone: 888-529-5929
Web: https://www.citiprogram.org

COLLABORATIVE INSTITUTIONAL TRAINING INITIATIVE (CITI PROGRAM)

COMPLETION REPORT - PART 2 OF 2
COURSEWORK TRANSCRIPT**

** NOTE: Scores on this Transcript Report reflect the most current quiz completions, including quizzes on optional (supplemental) elements of the course. See list below for details. See separate Requirements Report for the reported scores at the time all requirements for the course were met.

- **Name:** shamsul masum (ID: 6207588)
- **Institution Affiliation:** Massachusetts Institute of Technology Affiliates (ID: 1912)
- **Institution Email:** up642495@myport.ac.uk
- **Institution Unit:** school of engineering

- **Curriculum Group:** Human Research
- **Course Learner Group:** Data or Specimens Only Research
- **Stage:** Stage 1 - Basic Course

- **Record ID:** 22496305
- **Report Date:** 04-Mar-2017
- **Current Score**:** 92

REQUIRED, ELECTIVE, AND SUPPLEMENTAL MODULES	MOST RECENT	SCORE
History and Ethics of Human Subjects Research (ID: 498)	02-Mar-2017	6/7 (86%)
Belmont Report and CITI Course Introduction (ID: 1127)	02-Mar-2017	3/3 (100%)
Records-Based Research (ID: 5)	02-Mar-2017	3/3 (100%)
Genetic Research in Human Populations (ID: 6)	02-Mar-2017	4/5 (80%)
Research and HIPAA Privacy Protections (ID: 14)	04-Mar-2017	4/5 (80%)
Conflicts of Interest in Research Involving Human Subjects (ID: 488)	04-Mar-2017	5/5 (100%)
Basic Institutional Review Board (IRB) Regulations and Review Process (ID: 2)	02-Mar-2017	5/5 (100%)
Populations in Research Requiring Additional Considerations and/or Protections (ID: 16680)	04-Mar-2017	5/5 (100%)
Massachusetts Institute of Technology (ID: 1290)	04-Mar-2017	No Quiz

For this Report to be valid, the learner identified above must have had a valid affiliation with the CITI Program subscribing institution identified above or have been a paid Independent Learner.

Verify at: www.citiprogram.org/verify/?k3275c93f-e1f9-40fa-a3a5-47f6647ca7bb-22496305

Collaborative Institutional Training Initiative (CITI Program)
Email: support@citiprogram.org
Phone: 888-529-5929
Web: https://www.citiprogram.org

Appendix C

BP Forecast performance using LSTM algorithms and different strategies

Table C.1: RMS Error of forecast models with LSTM algorithm and different forecasting strategies for forecasting 30 min of BP

Patient ID	Recursive	Direct	MIMO	DirRec	DIRMO	RECMO
1	6.028	6.028	10.808	6.108	5.339	7.294
2	9.593	8.763	7.997	11.244	8.879	7.329
3	12.34	12.834	4.174	11.261	5.293	8.118
4	9.878	9.994	4.927	15.713	7.539	6.263
5	17.142	3.45	4.331	8.775	4.521	3.832
6	8.846	8.603	5.026	8.022	3.299	4.8
7	7.157	8.173	7.891	17.034	27.079	9.909
8	18.332	13.882	9.762	14.751	7.318	6.466
9	2.477	5.441	1.885	3.054	2.806	2.238
10	8.245	7.568	5.641	7.791	6.093	7.971
11	21.249	12.846	6.055	15.914	9.905	14.311
12	37.179	35.731	17.291	25.26	11.213	21.961
13	18.385	15.981	13.222	15.239	17.713	12.611
14	2.516	2.462	10.166	2.406	37.219	40.113
15	9.153	9.337	9.567	9.475	9.656	9.486
16	7.556	7.468	2.735	9.275	3.675	2.826
17	6.566	15.18	7.118	16.437	4.615	5.608
18	9.923	36.089	18.689	6.726	9.543	11.795
19	10.908	12.586	9.356	12.164	13.696	8.323
20	10.601	9.321	52.077	9.454	10.725	10.281
21	7.968	10.717	7.256	7.137	10.561	6.869
22	21.214	19.898	10.373	20.232	12.034	14.422
23	3.507	4.346	5.013	4.019	3.882	4.198
24	8.906	9.249	8.082	8.935	8.237	8.42
25	30.536	31.071	16.074	32.276	11.568	20.392
26	19.382	18.098	20.117	18.208	12.244	15.437
27	8.547	8.437	6.426	7.897	6.1	7.193
28	8.28	8.896	8.677	41.705	6	10.165
29	0.65	0.708	0.744	0.64	0.871	0.917
30	4.111	4.325	5.132	4.616	5.151	5.445

Appendix D

BP Forecast performance using BI-LSTM algorithms and different strategies

Table D.1: RMS Error of forecast models with BI-LSTM algorithm and different forecasting strategies for forecasting 30 min of BP

Patient ID	Recursive	Direct	MIMO	DirRec	DIRMO	RECMO
1	5.139	5.027	7.915	5.657	9.188	7.397
2	10.631	10.973	7.999	10.824	8.073	6.636
3	12.132	12.506	5.588	12.132	4.98	8.753
4	13.591	13.591	5.078	13.887	5.396	5.794
5	12.367	13.182	6.008	10.15	4.529	5.542
6	7.923	7.171	4.713	7.257	4.06	4.985
7	7.193	8.82	8.142	6.711	7.821	8.337
8	17.466	19.877	6.55	16.879	6.024	6.618
9	5.153	3.871	1.755	4.904	1.536	1.497
10	15.737	12.866	5.629	14.612	3.904	6.038
11	16.142	13.157	5.92	13.702	5.008	9.11
12	33.597	33.921	13.441	34.916	7.77	17.148
13	16.076	16.84	9.746	17.387	7.3	11.333
14	2.403	2.512	2.434	2.339	2.426	2.435
15	8.875	8.822	9.566	9.048	9.652	9.573
16	6.631	8.516	3.074	6.266	3.197	3.139
17	41.501	14.022	5.929	19.422	5.195	5.269
18	6.582	8.369	12.5	8.387	5.374	12.575
19	11.246	11.455	11.539	11.207	9.781	8.549
20	9.41	9.398	11.745	9.648	12.047	11.187
21	8.128	6.375	5.905	6.947	6.056	8.346
22	21.038	20.184	10.861	21.038	12.179	14.644
23	3.127	3.142	3.979	3.445	2.893	3.743
24	9.232	9.177	8.082	9.479	7.639	8.73
25	30.269	31.216	18.272	30.846	10.618	18.536
26	19.333	19.404	15.872	18.179	12.967	15.319
27	9.893	8.477	6.426	8.551	5.563	6.6
28	6.265	7.003	6.021	6.641	5.45	8.438
29	0.6	0.602	0.811	0.607	0.752	0.742
30	4.027	3.981	5.721	3.985	5.497	4.874

Appendix E

HR Forecast performance using LSTM algorithms and different strategies

Table E.1: RMS Error of forecast models with LSTM algorithm and different forecasting strategies for forecasting 30 min of HR

Patient ID	Recursive	Direct	MIMO	DirRec	DIRMO	RECMO
1	8.848	3.344	5.045	4.662	5.164	4.977
2	0.023	0.068	1.816	0.196	1.678	0.687
3	6.183	7.044	4.12	6.594	3.938	5.427
4	1.61	1.598	1.657	1.613	1.623	1.631
5	5.206	4.764	0.423	4.416	2.455	1.542
6	6.682	6.166	5.512	6.272	5.665	4.383
7	9.217	8.652	6.119	7.915	5.045	6.396
8	4.897	6.166	1.014	9.565	3.376	3.12
9	47.05	8.361	9.128	42.137	7.514	7.091
10	12.166	11.491	12.321	4.622	3.94	9.178
11	3.093	2.756	8.317	3.444	5.281	2.982
12	12.569	9.241	1.17	12.582	5.839	6.75
13	30.266	27.394	25.738	30.879	25.267	24.752
14	1.751	2.469	1.133	2.138	3.4	1.985
15	1.706	1.748	1.382	3.074	1.498	2.355
16	10.154	9.331	5.15	10.631	3.662	5.524
17	8.536	9.675	4.845	10.959	7.019	6.522
18	8.656	9.646	10.981	6.729	9.476	7.304
19	19.995	13.818	7.403	8.301	5.942	4.887
20	6.534	8.952	5.049	9.132	2.825	4.472
21	12.65	14.693	11.552	13.871	12.771	12.276
22	13.12	13.713	4.126	18.576	5.021	3.5
23	10.82	10.539	6.025	10.289	6.671	7.232
24	5.424	4.694	3.834	5.393	3.268	4.286
25	3.714	3.815	1.443	3.934	0.669	3.257
26	7.196	5.719	2.36	3.808	3.794	2.393
27	4.081	5.948	0.57	3.929	2.839	4.834
28	5.556	10.512	1.48	6.642	2.187	8.233
29	1.641	2.812	1.798	2.17	1.5	2.162
30	3.426	4.086	3.337	3.057	3.82	2.686

Appendix F

HR Forecast performance using BI-LSTM algorithms and different strategies

Table F.1: RMS Error of forecast models with BI-LSTM algorithm and different forecasting strategies for forecasting 30 min of HR

Patient ID	Recursive	Direct	MIMO	DirRec	DIRMO	RECMO
1	2.569	17.561	3.951	2.428	4.003	3.636
2	0.031	0.009	0.004	0.016	0.004	0.013
3	6.698	6.669	4.286	7.043	4.036	5.362
4	1.619	1.612	1.651	1.613	1.644	1.647
5	5.755	4.383	3.087	5.13	4.148	3.252
6	6.302	6.323	4.132	6.431	2.026	3.956
7	9.191	8.808	6.118	9.924	5.206	6.481
8	0.274	0.26	0.107	0.251	0.217	0.266
9	7.153	4.589	12.598	23.562	5.752	4.608
10	8.785	8.669	5.731	7.816	2.331	4.925
11	2.683	2.712	4.061	2.614	4.596	2.881
12	19.351	14.963	2.981	14.299	2.99	8.325
13	32.36	29.305	25.185	31.646	25.377	25.309
14	1.726	1.378	1.704	1.705	1.512	1.673
15	1.319	1.381	1.582	1.554	1.247	1.447
16	5.844	6.299	5.69	6.301	2.522	4.429
17	8.03	10.045	4.901	9.136	5.619	10.466
18	7.283	6.97	7.28	7.293	7.104	7.345
19	13.29	12.835	4.449	15.79	4.158	4.266
20	2.645	4	1.929	3.526	2.201	3.899
21	15.475	13.493	16.002	14.969	19.281	15.724
22	7.505	8.537	2.844	8.265	1.527	4.419
23	9.535	9.704	6.856	9.675	6.805	7.077
24	4.969	4.77	3.231	5.347	3.055	3.831
25	2.783	3.278	1.906	3.992	1.575	1.309
26	5.833	4.162	2.746	5.86	1.801	1.057
27	21.532	19.58	0.986	24.444	0.332	0.461
28	0.703	2.231	1.519	0.829	1.159	0.957
29	1.65	2.081	1.25	1.342	1.281	2.079
30	5.362	5.694	3.53	3.242	4.221	2.96

Appendix G

Forecast performance of CNN model

Table G.1: Forecast performance of CNN model

PATIENT ID	CNN			
	MIMO		DIRMO	
	UNIVARIATE	MULTIVARIATE	UNIVARIATE	MULTIVARIATE
1	8.407	8.77	8.648	9.725
2	7.913	8.201	8.973	8.195
3	4.53	5.657	4.671	6.133
4	4.622	4.42	4.872	4.848
5	4.655	4.933	4.988	4.303
6	6.98	6.125	6.34	6
7	8.589	7.523	8.452	8.019
8	7.13	7.28	7.114	6.725
9	1.531	1.888	1.603	2.789
10	3.567	2.905	3.267	3.056
11	6.447	5.593	8.011	6.867
12	15.396	8.881	12.634	3.509
13	13.673	9.292	12.852	8.751
14	3.416	2.743	3.226	3.104
15	9.668	9.521	9.876	9.832
16	10.111	9.689	9.995	9.061
17	2.607	4.11	3.868	6.195
18	5.7	6.55	5.817	5.743
19	9.419	10.019	9.133	5.697
20	9.534	10.425	9.715	10.595
21	13.957	10.426	13.645	12.061
22	7.722	7.893	7.341	6.639
23	5.087	3.928	4.333	3.513
24	8.068	8.293	8.024	7.695
25	12.689	11.38	11.002	6.687
26	10.323	10.069	10.767	10.467
27	6.051	6.326	5.636	5.741
28	6.874	8.506	7.307	6.803
29	0.77	0.697	0.979	1.017
30	6.155	7.547	6.232	6.335

Appendix H

Forecast performance of LSTM model

Table H.1: Forecast performance of LSTM model

PATIENT ID	LSTM			
	MIMO		DIRMO	
	UNIVARIATE	MULTIVARIATE	UNIVARIATE	MULTIVARIATE
1	6.359	7.561	9.791	9.87
2	12.426	9.415	8.927	8.742
3	3.727	4.934	5.768	5.816
4	6.89	6.241	5.956	6.481
5	2.831	2.481	4.322	3.789
6	6.513	6.3	6.301	6.109
7	13.587	8.015	9.233	8.674
8	7.303	6.935	6.286	6.372
9	1.598	2.117	1.569	1.909
10	3.312	3.29	3.194	3.012
11	8.105	8.883	6.913	6.32
12	9.921	7.335	12.409	5.843
13	15.716	12.368	9.382	6.534
14	6.742	9.555	4.334	7.645
15	9.771	9.518	9.403	9.491
16	9.998	9.519	9.348	9.125
17	4.486	2.851	3.976	3.306
18	6.875	5.308	6.781	5.267
19	13.027	16.149	18.001	19.86
20	14.375	11.55	10.828	10.852
21	13.839	10.925	17.477	14.636
22	9.892	11.418	9.581	6.386
23	6.889	6.349	4.833	5.802
24	7.497	7.425	7.676	7.699
25	14.82	11.966	9.77	8.498
26	9.347	9.19	9.839	9.854
27	10.795	9.716	8.972	10.057
28	16.042	13.835	10.774	10.639
29	1.868	1.471	1.268	0.764
30	6.829	6.57	11.779	6.728

Appendix I

Forecast performance of BI-LSTM model

Table I.1: Forecast performance of BI-LSTM model

PATIENT ID	BI-LSTM			
	MIMO		DIRMO	
	UNIVARIATE	MULTIVARIATE	UNIVARIATE	MULTIVARIATE
1	8.218	7.763	9.744	8.787
2	8.345	7.945	8.709	8.453
3	4.38	4.5	5.539	5.605
4	5.985	5.617	6.036	5.532
5	4.152	4.218	4.204	4.443
6	5.789	5.671	5.529	5.432
7	7.877	7.335	8.027	7.65
8	6.194	6.122	5.75	6.197
9	2.353	1.719	1.706	1.61
10	3.219	3.11	3.001	2.899
11	5.757	5.292	7.312	8.514
12	10.385	7.729	5.921	4.593
13	12.357	11.171	9.154	9.828
14	4.954	5.472	4.541	4.144
15	9.63	9.577	9.634	9.596
16	9.5	8.906	8.809	8.498
17	2.978	3.051	3.917	4.046
18	7.636	5.004	7.127	4.855
19	6.158	6.91	5.302	5.356
20	11.739	12.346	12.28	11.216
21	12.938	10.884	13.367	11.691
22	7.385	7.295	6.635	6.234
23	4.862	4.907	3.782	3.552
24	7.334	7.498	7.572	7.914
25	9.853	9.293	7.852	6.637
26	9.175	10.214	9.405	9.602
27	7.091	7.002	6.409	6.75
28	9.497	11.977	9.186	7.785
29	0.663	0.86	0.679	0.832
30	7.465	6.797	6.971	5.398

Appendix J

Error analysis in different forecast horizon

Table J.1: Error analysis in different forecast horizon

Patients	Forecast Horizon											
	10	20	30	40	50	60	70	80	90	100	110	120
1	3.918	4.278	8.787	5.283	4.883	3.509	4.114	4.495	4.969	5.423	5.266	5.207
2	3.329	9.447	8.453	6.327	5.533	7.708	8.412	9.563	9.027	10.954	11.267	12.86
3	6.664	5.888	5.605	5.748	10.4	9.72	10.577	11.269	11.621	12.336	12.872	13.35
4	8.248	12.302	5.532	7.102	7.314	7.528	8.803	9.028	9.263	9.482	10.809	11.447
5	1.827	2.666	4.443	4.593	2.994	2.095	2.665	2.8	3.398	5.197	7.892	8.876
6	6.809	9.008	7.65	9.179	4.652	4.403	6.206	6.541	6.634	6.895	10.979	12.574
7	4.873	5.277	6.197	9.901	10.067	10.537	11.022	12.021	12.344	13.04	13.091	13.99
8	1.721	2.598	1.61	2.336	3.669	3.943	6.425	7.667	8.789	9.899	10.921	12.566
9	7.749	3.588	8.514	20.585	19.692	18.568	20.512	17.098	15.74	17.527	16.508	17.098
10	2.642	3.475	4.593	16.36	25.009	14.75	18.561	17.074	16.912	18.571	19.643	20.983
11	4.104	15.488	9.828	10.743	8.334	10.027	12.253	14.728	16.254	16.022	15.968	16.227
12	2.975	6.509	4.144	2.708	3.154	3.204	3.615	4.068	4.685	6.046	7.629	17.793
13	5.623	7.325	9.596	7.684	2.981	4.147	5.798	6.56	6.82	7.43	8.445	9.699
14	3.107	3.98	4.046	5.474	6.707	6.477	7.028	8.438	9.48	10.35	11.777	12.908
15	2.223	7.93	4.855	4.028	9.023	6.46	7.669	7.876	8.639	11.266	11.889	12.655
16	6.759	11.643	5.356	6.374	13.099	15.139	16.589	17.25	17.99	18.349	19.377	20.132
17	4.827	13.622	11.216	8.92	9.003	7.505	8.951	9.09	10.59	11.504	13.842	15.881
18	3.344	10.387	11.691	10.94	9.165	7.905	8.122	8.267	10.378	11.663	12.271	12.333
19	3.542	2.245	6.234	8.191	7.297	4.845	8.235	9.276	9.99	10.903	12.59	13.954
20	5.648	4.997	3.552	5.721	5.262	5.03	6.789	7.904	10.961	12.981	14.25	15.609
21	18.444	12.245	7.914	4.863	6.003	4.536	7.201	8.945	9.912	10.43	12.167	14.951
22	3.252	6.891	6.637	12.19	12.406	10.511	10.7	10.9	10.969	14.385	12.181	11.237
23	18.857	12.003	9.602	7.626	13.104	16.981	17.963	18.188	19.479	20.27	21.721	22.265
24	12.627	9.668	6.75	5.461	4.108	3.023	5.209	7.541	8.934	10.012	13.678	15.49
25	9.491	10.805	7.785	14.325	14.076	13.341	15.951	16.527	17.532	17.901	18.741	19.051
26	0.913	0.565	0.832	1.893	2.015	2.876	3.324	3.925	4.36	4.69	5.908	6.739
27	3.603	5.481	5.098	4.897	3.809	5.235	6.255	7.499	7.651	7.817	8.623	9.169
28	3.239	3.654	4.139	5.696	6.698	6.588	7.135	8.641	9.51	9.85	11.555	12.89
29	3.373	6.753	6.57	12.21	12.39	11.01	11.8	11.95	11.989	12.475	12.9	13.27
30	3.633	5.081	5.398	4.197	3.889	5.682	6.133	7.303	7.559	7.928	8.512	9.057

Appendix K

Research Ethics Review Checklist

FORM UPR16

Research Ethics Review Checklist

Please include this completed form as an appendix to your thesis (see the Research Degrees Operational Handbook for more information

Postgraduate Research Student (PGRS) Information							
Student ID:	642495						
PGRS Name:	Shamsul Masum						
Department:	SENE	**First Supervisor:**	Dr John Chiverton				
Start Date: (or progression date for Prof Doc students)	**October, 2015**						
Study Mode and Route:	Part-time ☐	MPhil ☐	MD ☐				
	Full-time ☒	PhD ☒	Professional Doctorate ☐				

Title of Thesis:	Forecasting from Physiological Time Series Through Supervised Learning
Thesis Word Count: (excluding ancillary data)	~ 54407

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:
(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES ☒ NO ☐
b) Have all contributions to knowledge been acknowledged?	YES ☒ NO ☐
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES ☒ NO ☐
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES ☒ NO ☐
e) Does your research comply with all legal, ethical, and contractual requirements?	YES ☒ NO ☐

Candidate Statement:

I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)

Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	A840-B4B9-7BBF-27D5-4552-9B63-164A-11E3

If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:

Signed *(PGRS)*:		**Date:** 30-03-2021